

(NASA-TM-X-56038) A FORTRAN PROGRAM FOR THE
ANALYSIS OF LINEAR CONTINUOUS AND
SAMPLED-DATA SYSTEMS (NASA)

N76-18823

Unclassified
G3/61 18568

NASA TM X-56038

A FORTRAN PROGRAM FOR THE ANALYSIS OF LINEAR
CONTINUOUS AND SAMPLED-DATA SYSTEMS

John W. Edwards

January 1976

NASA high-number Technical Memorandums are issued to provide rapid transmittal of technical information from the researcher to the user. As such, they are not subject to the usual NASA review process.

NASA Dryden Flight Research Center
Edwards, California 93523

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

1

NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED
FROM THE BEST COPY FURNISHED US BY
THE SPONSORING AGENCY. ALTHOUGH IT
IS RECOGNIZED THAT CERTAIN PORTIONS
ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE
AS MUCH INFORMATION AS POSSIBLE.

1. Report No. TM X-56038	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A FORTRAN PROGRAM FOR THE ANALYSIS OF LINEAR CONTINUOUS AND SAMPLED-DATA SYSTEMS		5. Report Date January 1976
7. Author(s) John W. Edwards		6. Performing Organization Code
9. Performing Organization Name and Address NASA Dryden Flight Research Center P. O. Box 273 Edwards, California 93523		8. Performing Organization Report No
		10. Work Unit No. 505-06-91
		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		
16. Abstract		
<p style="text-align: center;">A FORTRAN digital computer program which performs the general analysis of linearized control systems is described. State variable techniques are used to analyze continuous, discrete, and sampled-data systems. Analysis options include the calculation of system eigenvalues, transfer functions, root loci, root contours, frequency responses, power spectra, and transient responses for open- and closed-loop systems. A flexible data input format allows the user to define systems in a variety of representations. Data may be entered by inputting explicit data matrices or matrices constructed in user-written subroutines, by specifying transfer function block diagrams, or by using a combination of these methods.</p>		
<p style="text-align: center;">ORIGINAL PAGE IS OF POOR QUALITY</p>		
17. Key Words (Suggested by Author(s)) FORTRAN linear analysis program Continuous and sampled data analysis Control system analysis		18. Distribution Statement Unclassified - Unlimited
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified

*For sale by the National Technical Information Service, Springfield, Virginia 22151



CONTENTS

	Page
INTRODUCTION	1
SYMBOLS	1
PROGRAM DESCRIPTION	3
CONTINUOUS SYSTEM MODELS	7
Open Loop	7
Closed Loop	7
Root Loci	8
MIXED Systems	10
DATA DECK FORMAT	11
DATA INPUT OPTIONS	12
CHANGE	13
LOAD	13
MATRIX	14
CLASS	14
MIXED INPUT OPTION (CONTINUOUS SYSTEMS)	27
DISCRETE SYSTEM MODELS	40
SAMPLED-DATA SYSTEM MODELS	41
SAMPLED-DATA SYSTEM ANALYSIS USING THE MIXED OPTION	58
COMPUTATIONAL DELAY AND MODIFIED z-TRANSFORMS	64
METHOD OF ANALYSIS	68
Frequency Response and Power Spectra	68
Eigenvalues	70
Transfer Function Numerators	70
Transient Response and Discretization of Sampled-Data Systems	70
	Pages ii - iv BLANK

	Page
Digital Filtering	72
Model Following	74
Similarity Transformation	74
PROGRAM SIZE AND TIMING	75
EXAMPLE PROBLEM	77
CONCLUDING REMARKS	79
REFERENCES	80
APPENDIX 1: FORTRAN LISTING OF CONTROL PROGRAM	81
APPENDIX 2: NAMELIST CODE	188
APPENDIX 3: INPUT AND OUTPUT LISTINGS FOR EXAMPLE	192
TABLE I: CONTROL PROGRAM ANALYSIS OPTIONS	211
TABLE II: CONTROL PROGRAM JOB CONTROL LANGUAGE AS OF 11-1-74	212
TABLE III: CONTROL: SYSTEM MODELS	216
TABLE IV: CONTROL PROGRAM DATA DECK FORMAT	218
TABLE V: DATA REQUIRED BY CONDITION CODES	219
TABLE VI: TRANSFER FUNCTION STANDARD FORMS (NIT = 1)	220
TABLE VII: DISCRETIZED PLANT MODELS FOR VARIOUS INPUT AND OUTPUT DEFINITIONS	222
TABLE VIII: VECTOR ORDERING AND SYSTEM CONNECTION CONVENTIONS FOR SAMPLED-DATA SYSTEMS	223
FIGURES	224

A FORTRAN PROGRAM FOR THE ANALYSIS
OF LINEAR CONTINUOUS AND SAMPLED-DATA SYSTEMS

John W. Edwards
Dryden Flight Research Center

INTRODUCTION

A FORTRAN digital computer program is described which analyzes linear continuous or sampled-data systems using state variable techniques. Open- and closed-loop systems may be analyzed using frequency response or transient response techniques. Root locus and root contour options are also available. Systems may be defined by inputting explicit data matrices, by constructing matrices in user written subroutines, or by specifying transfer function block diagrams. The program also allows the user to define a system using a combination of the above methods. For instance, the plant may be described as a set of coupled differential equations and its control system described by a block diagram. The program allows the user to analyze such a system without converting the subsystems into a common representation.

SYMBOLS

$A, B, C, H, G, F,$ K_1, K_2, K_3, K_4, D	matrices used to define system models
$G(\cdot)$	transfer function
I	identity matrix
m	fractional computational time delay for sampled-data systems

N_X, N_Y, N_U	dimensions of state, output, and input vectors
R, S	matrices used to define system inter- connections for MIXED option
S, \mathfrak{Z}, W	Laplace, \mathfrak{Z} - , and W - transform vari- ables
\mathfrak{Z}, V	output and input vectors for block diagram portions of MIXED systems
X, Y, U	state, output, and input vectors
t	time-sec.
T	sampled-data system sample period-sec.
ϵ	incremental time period-sec.
$\phi(t)$	state transition matrix
$\Theta(t)$	integral of $\phi(t)$
Subscripts:	
com	command signal
ext	external signal
n	time index for sampled-data systems
Superscripts:	
c	designates continuous subsystem of sampled-data system
d	designates discrete subsystem of sampled-data system
T	transpose of a matrix

PROGRAM DESCRIPTION

CONTROL is a FORTRAN digital program capable of performing general analysis of linearized control systems problems. It utilizes state variable matrix operations to find system eigenvalues, transfer functions, root contours, root loci, frequency responses, power spectra, and transient responses. Continuous, discrete, and sampled-data system analyses may be accomplished. The data input format is quite flexible, allowing data to be entered in matrix form, block diagram form, general parameter input form, or a combination of these forms. The analysis options available are listed in Table I.

The basic systems which CONTROL analyses are

$$\dot{x} = Ax + Bu \quad (\text{plant equation}) \quad (1a)$$

$$y = Hx + Fu \quad (\text{output equation}) \quad (1b)$$

$$u = K_1 x + Du_{\text{com}} \quad (\text{control law}) \quad (1c)$$

for continuous systems and

$$x_{n+1} = Ax_n + Bu_n \quad (\text{plant equation}) \quad (2a)$$

$$y_n = Hx_n + Fu_n \quad (\text{output equation}) \quad (2b)$$

$$u_n = K_1 x_n + Du_{\text{com}_n} \quad (\text{control law}) \quad (2c)$$

for discrete systems. The state vector, x , is N_X dimensional; the output vector, y , is N_Y dimensional; and the input vector, u , is N_U dimensional.

To allow user flexibility, additional matrix equations are used in the program for definition of feedback control laws, etc., and will be discussed later. In all cases, the

system equations are reduced to the form of systems (1) or (2) before being analyzed.

CONTROL program analysis is performed in the CNTRLR subroutine which is diagrammed in figure 1. The use of each subroutine is briefly described below.

CARD - reads data cards which specify program options chosen

LOAD, MATRIX, CHANGE, CLASS - data input subroutines

LOAD - reads data input matrices

MATRIX - a user written subprogram which constructs system
matrices from basic data

CHANGE - a user written subprogram which changes data already
in the program. Used for parameter variation studies

CLASS - constructs system matrices from data describing a
block diagram of transfer functions

SETUP - reduces input data matrices to the system equations
(1) or (2)

EIGEN - determines system eigenvalues from the A matrix using
QR-algorithm

ROOT - performs root loci analysis as a function of two inde-
pendently incremented feedback gains

NMRATR - determines numerator zeroes of transfer functions
defined by the NU inputs, u, and NY outputs, y.

FRQRSP - computes a frequency response at discrete frequen-
cies for each transfer function determined by NMRATR

PSD - computes a power spectrum of the transfer functions
determined by NMRATR assuming unity variance white
noise excitation

THIST - computes a transient response of the system at discrete time points

INPUT - a user written subroutine which constructs the input vector to the transient response

The CONTROL program is on disc in the CDC CYBER 70 system and is called by the user with appropriate control cards. To utilize the full flexibility of CONTROL, the user may write several of the subroutines to perform data preparation for his specific problem. These subroutines are:

MAIN

MATRIX

CHANGE

INPUT

These subroutines may be compiled in SOURCE language.

Note that they are not required. These subroutines are defined in the program on disc and are overridden by the user's source subroutines. If a specific output format is desired, any of the CONTROL subroutines may be modified by the user and compiled to override the disc versions. A brief description of the use of these routines is given below:

MAIN - Provides variable dimensioning capability. The size of all arrays used in CONTROL is declared here and passed through COMMON to all other subroutines. Thus the size of the system matrixes may be quickly and easily changed. The disc program is provided with:

MX = 15 = n + 1; n = dimension of maximum state

MY = 15 = dimension of maximum output vector

MY = 10 = dimension of maximum input vector

MATRIX - A user written subroutine which constructs the system matrices. For instance, MATRIX may read the nondimensional aircraft stability derivatives, perform axes transformations, dimensionalize the derivatives, and insert the proper numbers into the system matrixes. General MATRIX routines are available for the standard lateral-directional and longitudinal linearized equations of motion which are generally used at FRC.

CHANGE - A user written subroutine which changes specified elements in the system matrices (which are already constructed through MATRIX, LOAD, or CLASS). This routine allows the capability of doing parameter variation studies on a basic system configuration without having to reload the problem data for each variation.

INPUT - This user written routine constructs the input vector for a time history calculation. Thus the user may generate step, impulse, ramp, sinusoidal, random inputs as desired. A basic routine which simply reads 1 card for the input vector u is provided.

All of these routines have specified calling sequences which are given in Appendix 1. Each routine has specific COMMON

and DIMENSION statements which are required and which the user will not change. The structure of the CONTROL deck is shown in figure 2. The job control language for CONTROL is given in Table II. All figures, tables and information required to set up data for the CONTROL program are gathered at the end of this document for easy reference by the user.

CONTINUOUS SYSTEM MODELS

The basic continuous system model which CONTROL analyzes is given by (1) and is repeated here:

$$\dot{x} = Ax + Bu$$

$$y = Hx + Fu$$

$$u = K_1x + D_{u\text{com}}$$

CONTROL analyzes three configurations of continuous system models: open loop, closed loop, and root loci. These three configurations are under the control of SYSTEM. For ease in problem setup, the additional matrices D, K₁, K₂, K₃, K₄, C, and G are defined.

Open Loop (SYSTEM = 1)

$$C\dot{x} = Ax + Bu \quad (3a)$$

$$y = Hx + G\dot{x} + Fu \quad (3b)$$

Closed Loop (SYSTEM = 2)

$$C\dot{x} = Ax + Bu \quad (4a)$$

$$u = K_1x + K_2\dot{x} + Du_{\text{com}} \quad (4b)$$

$$y = Hx + G\dot{x} + Fu \quad (4c)$$

Subscript COM indicates a "command" signal. Note that the

inclusion of the C matrix allows inertial cross-coupling between states. K2 and G allow for state derivative feedback and output. D allows for controller interconnections and feedforward gains.

CONTROL reduces the systems (3) and (4) with the substitutions

$$A \leftarrow C'A + C'B(I - K_2 C'B)^{-1}(K_1 + K_2 C'A)$$

$$B \leftarrow C'B(I - K_2 C'B)^{-1}D$$

$$H \leftarrow H + GC'A + (F + GC'B)(I - K_2 C'B)^{-1}D$$

$$F \leftarrow (F + GC'B)(I - K_2 C'B)^{-1}D$$

to the system,

$$\dot{x} = Ax + Bu_{\text{com}} \quad (5a)$$

$$y = Hx + Fu_{\text{com}} \quad (5b)$$

Note that (5) is identical to the basic system (1) with u replaced by u_{COM} . Obviously, for the open-loop system $D = I$, $K_1 = K_2 = 0$, and $u = u_{\text{COM}}$.

Root Loci (SYSTEM = 3)

$$C\dot{x} = Ax + Bu \quad (6a)$$

$$u = k_1(K_1x + K_2\dot{x}) + k_3(K_3x + K_4\dot{x}) \quad (6b)$$

This root loci model allows root loci as a function of two independent feedback variables. The first feedback variable is defined by the matrices K_1 and K_2 (e.g., normal acceleration $= \frac{V}{g}(\ddot{a} - q)$). K_3 and K_4 define the second feedback variable (commonly $K_3 = K_4 = 0$). The condition codes N_1 , N_2 , GAIN1, and GAIN2 determine the locus gain and number of locus points as follows:

$$k_1 = r * GAIN_1 \quad r = \begin{cases} 0, 1, 2, 3, \dots, N_1; & N_1 > 0 \\ 0, 1, 2, 4, \dots, 2^{\lfloor N_1/2 \rfloor}; & N_1 < 0 \end{cases}$$

$$k_3 = t * GAIN_2 \quad t = \begin{cases} 0, 1, 2, 3, \dots, N_2; & N_2 > 0 \\ 0, 1, 2, 4, \dots, 2^{\lfloor N_2/2 \rfloor}; & N_2 < 0 \end{cases}$$

CONTROL computes the closed-loop system matrix as:

$$\dot{x} = Ax$$

where

$$A \leftarrow A' + B' (I - k_1 K' - k_3 K^2)^{-1} (k_1 K^3 + k_3 K^4)$$

$$A' \leftarrow C^{-1}A$$

$$B' \leftarrow C^{-1}B$$

$$K' \leftarrow K_2 B'$$

$$K^2 \leftarrow K_4 B'$$

$$K^3 \leftarrow K_1 + K_2 A'$$

$$K^4 \leftarrow K_3 + K_4 A'$$

The eigenvalues of A are then determined to find the system root locus. The zeroes corresponding to the first feedback variable are determined by forming the observation equation:

$$y = Hx + Fu$$

where

$$H = \begin{bmatrix} K_1 + K_2 A' \\ K_3 + K_4 A' \end{bmatrix}$$

$$F = \begin{bmatrix} K_2 B' \\ K_4 B' \end{bmatrix}$$

A subsequent call to NMRATR then determines the zeroes.

NMRATR determines the zeroes of the transfer function specified by the first nonzero row of these derived H and F matrices. If PLOT #0, an additional data card is required for each root locus specifying the maximum and minimum axis coordinates desired for the plot (see Table II).

The continuous system models given above are summarized in Table III(a).

MIXED Systems

This problem formulation can be used to model a system which is described by a combination of differential equations and Laplace transform blocks. An example of such a system is an aircraft-control system. The aircraft equations of motion are usually known as differential equations while the control system is often given in block diagram form. The MIXED option gives the user the capability of easily modeling such a system. This option, which is a great convenience in analyzing continuous systems, becomes indispensable in the analysis of sampled-data systems. There it is used to correctly connect the continuous and discrete subsystems and to discretize to the continuous subsystem at the proper times.

The MIXED option involves a two-step data input stream. In the first step, the plant equations of motion are loaded as an open-loop system. The second step loads the block diagram control system data (CLASS subroutine) and augments the open-loop plant matrices with the control system dynamics.

Finally, the two subsystems are coupled together. The open-loop, closed-loop, and root locus options are available with the MIXED option.

The CLASS subroutine may be used by itself to construct state space formulations of systems described entirely by block diagrams. The input format for the MIXED option will be given after the CLASS loading option has been discussed.

DATA DECK FORMAT

CONTROL problem definition is accomplished in CARD. Condition codes defining the analysis options chosen, data input mode, and data handling procedures are read in a namelist format (CODE). Each pass through CNTRLR (figure 1) defines a case.

The CONTROL program data deck structure is shown in figure 3. The title card, namelist CODE, output label card, input label card, system data, and transient response input data formats are given in Table IV.

The format for entering data using namelist is illustrated in figure 4. Column 1 of each card must be blank. Column 2 of card 1 begins the name of the namelist (CODE) which is followed by a blank. Variable names in the namelist must be right justified with respect to the "=" sign, variable values must be right justified with respect to the "," sign and be of the proper type (real, integer). The namelist data entry is closed by the code ",&END."

The analysis options chosen, data input format, and system structure are defined in the namelist, CODE. The variables which are defined by the namelist are:

Integer variables

READ, SYSTEM, OUTPUT, MIXED, DIGITL, FRPS, NUMRS,
TRESP, NX, NY, NU, NXC, NUC, ZOH, N1, N2, CONTUR,
MULTRT, MODEL, NSCALE, CMAT, NK2, FORM, IPT, IGO,
SAV, IFLAG, READ3

Real variables

DELT, FINALT, IFREQ, FFREQ, DELFRQ, GAIN1, GAIN2, M

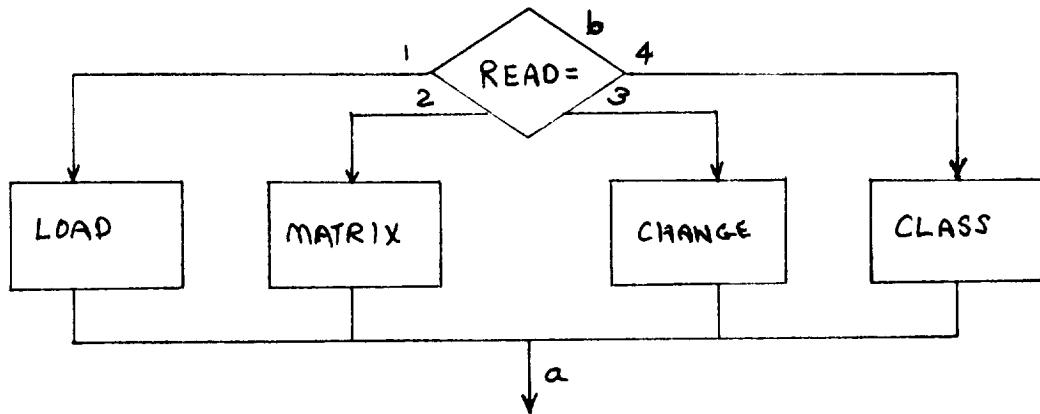
These variables are described in Appendix 2. All variables are initialized to zero at the start of each case (unless SAV = 1) and only nonzero variables need be defined in the namelist.

For root locus cases in which plots are requested, one card is required for each root locus by the plotter program to define the $j\omega$ scale on the plot. This card(s) follows the PLOT CARDS card.

Figure 1 indicates that CONTROL will return to the top of the program and look for more data at the end of each case. Thus data decks may be stacked together to analyze many cases on a single computer run.

DATA INPUT OPTIONS

The system data may be input in four different methods. The control loading option is determined by the condition code READ:



CHANGE (READ = 3) The CHANGE option is meant to be used on cases following an initial case. For instance, on case 1, the system matrices are loaded or constructed in LOAD, MATRIX, or CLASS. These "original" system matrices (as defined at point (a) on case 1) may be saved and made available at point (b) in case 2. If CHANGE is utilized in case 2, the "original" system matrices may be altered as desired. Thus a parameter variation study may be accomplished using a simple CHANGE subroutine.

The remaining three options--LOAD, MATRIX, and CLASS--will now be described.

LOAD (READ = 1) The subroutine LOAD reads explicit data matrices under the control of the condition codes. Each data matrix (e.g., A) is read row by row with an (8F10.4) format. Each data matrix must be preceded by a dimension card (2I10) giving the number of rows and columns in the matrix. For example, a 5 x 5 matrix would require 6 data cards (1 dimension card, 1 card per row) while an 8 x 9 matrix would require 19 data cards (1 dimension card, 2 cards per row). The matrices required are listed in STEP 1 of Table V.

MATRIX (READ = 2) MATRIX is a user written subroutine which constructs the system matrices required by the condition codes. These matrices are the same as those given in STEP 1 of Table V. Data defined in MATRIX are not destroyed by CONTROL. Thus the user may input data to the MATRIX subroutine for case 1 and reuse the data on subsequent cases.

CLASS (READ = 4) If READ = 4 the system matrices are constructed in CLASS from block diagram input data. If MIXED = 1, the control system dynamics, in block diagram form, are added on to the plant dynamics by the CLASS subroutine. Thus the CLASS subroutine has a dual function in CONTROL. In both cases, the data format describing the block diagram is the same. The data format is given as STEP 2 of Table V. Two options of inputting the block diagram information are provided. The first option describes the block diagram by inputting the transfer function polynomial coefficients explicitly for general fourth-order transfer functions. The second option allows the user to pick standard form constant, first-order, or second-order blocks to describe the system. These blocks are defined by a small number of parameters and are given in Table VI. The first option is used only if a particular transfer function form cannot be found in Table VI. The first option will not be described. A description of the second option begins on page 21.

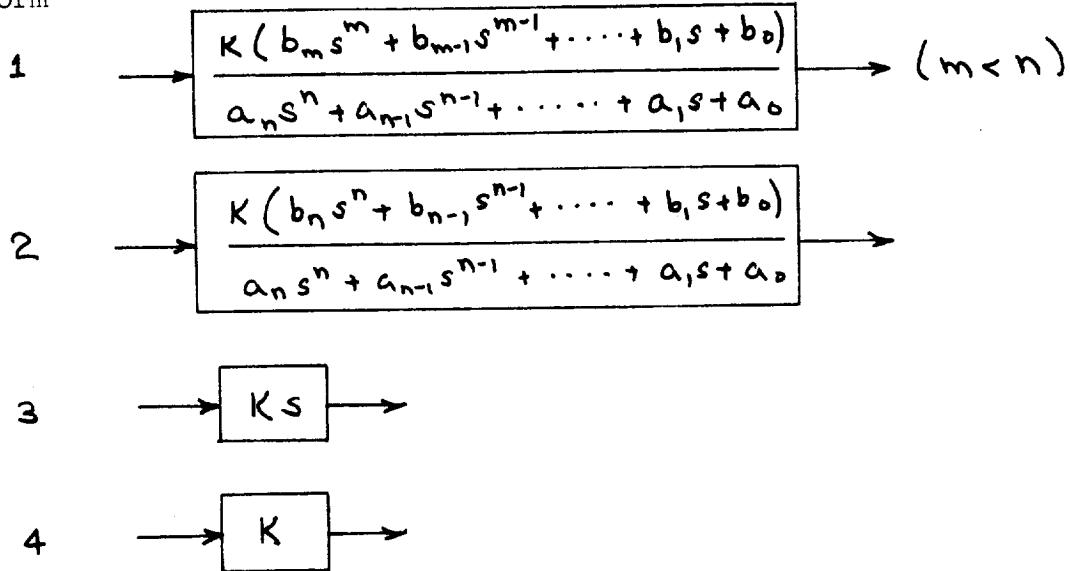
The data required to describe a system given entirely by block diagrams are summarized below. The additional data required if MIXED = 1 will be described later.

NBLOCK = No. of blocks in block diagram
GRAPH = Integer matrix describing interconnection of blocks
BLOCK = Integer matrix describing dimension of numerator
and denominator polynomials of each block
NUMER = Matrix of polynomial coefficients of numerators of
each block (ordered from $s^0 \rightarrow s^m$)
DENOM = Matrix of polynomial coefficients of denominators
of each block (ordered from $s^0 \rightarrow s^n$)
GAIN = Vector of gains of each block

ITHINY = Integer vector describing which block outputs are
to be saved in final H matrix

Allowable forms of blocks:

Form



Restrictions:

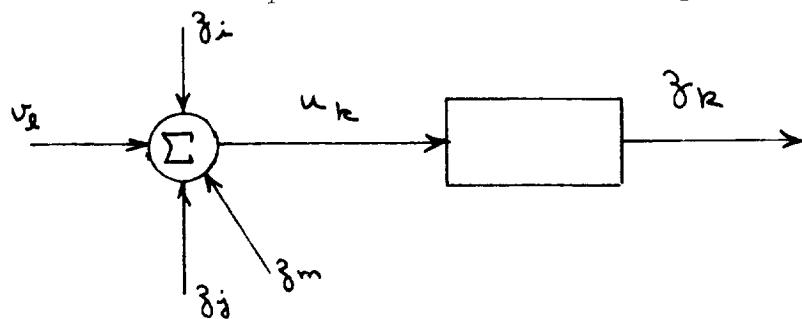
- a.) Inputs into a type 3.) block must be outputs of a type 1.) block.
- b.) $n \leq 4$
- c.) Number of blocks ≤ 20

Interconnection of blocks:

The interconnection of blocks may be described with the concept of internal and external inputs to a block:

Internal input - an input which is an output of a block in
the system.

External input - any input other than internal inputs, designated by v_j . The input to any block may be the sum of three or less internal inputs and one external input.



$$u_k = \pm v_k \pm z_i \pm z_j \pm z_m$$

The \pm indicates that either sign may be used on any of the inputs. Whenever an external input is defined, a summing junction is implied at the input to the block.

Problem formulation

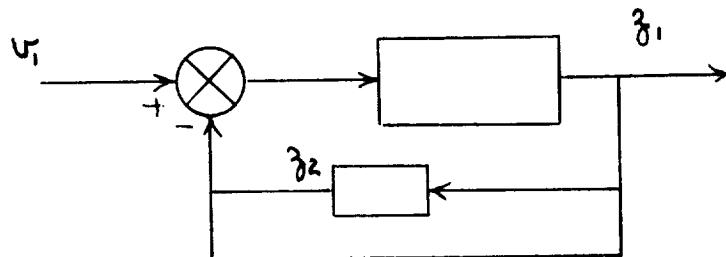
1. Label the outputs of the blocks consecutively z_i , $i = 1, 2, 3, \dots$
2. Label the external inputs consecutively v_j , $j = 1, 2, 3, \dots$
3. Construct the GRAPH Matrix (dimension NBLOCK x 5) where
NBLOCK = number of blocks in the diagram:

$$\text{GRAPH} = \left[\begin{array}{ccccc} | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \end{array} \right]$$

Annotations below the matrix:

- BLOCK NO.**: Points to the first column.
- INTERNAL INPUTS**: Points to the second through fourth columns.
- EXTERNAL INPUT**: Points to the fifth column.

A simple example will illustrate the method:



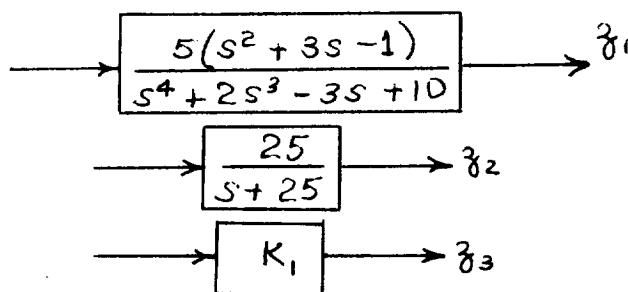
$$\text{GRAPH} = \begin{bmatrix} 1 & -2 & -1 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \end{bmatrix}$$

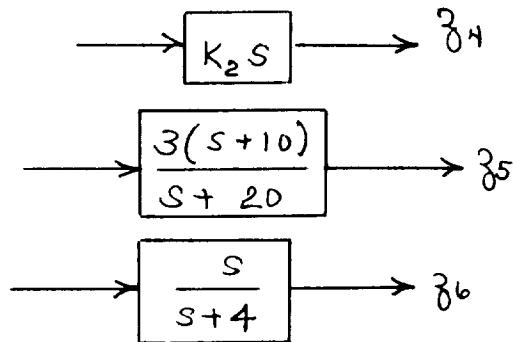
Note that negative inputs to summing junctions are indicated by the negative sign attached to the input label.

The following input data describe the internal composition of each block:

BLOCK - Describes the number of coefficients of the numerator and denominator polynomials which will be loaded (dimension NBLOCK x 3). The first column contains the block number (the sequence must correspond to the sequence of the first column of GRAPH). The second column contains the number of numerator coefficients to be loaded. The third column contains the number of denominator coefficients to be loaded.

example:





$$\text{BLOCK} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 1 & 2 \\ 3 & 1 & 1 \\ 4 & 2 & 1 \\ 5 & 2 & 2 \\ 6 & 2 & 2 \end{bmatrix}$$

Note that a constant block K is treated as $\frac{K}{1}$.

Also

$$Ks = \frac{K(s+0)}{1}$$

NUMER (NBLOCK x 5)

NUMER contains the numerator coefficients.

For the above example:

$$\text{NUMER} = \begin{bmatrix} -1. & 3. & 1. & 0 & 0 \\ 25. & 0 & 0 & 0 & 0 \\ 1. & 0 & 0 & 0 & 0 \\ 0 & 1. & 0 & 0 & 0 \\ 10. & 1. & 0 & 0 & 0 \\ 0 & 1. & 0 & 0 & 0 \end{bmatrix}$$

DENOM (NBLOCK x 5)

Denom contains the denominator coefficients.

For the above example:

$$\text{DENOM} = \begin{bmatrix} 10. & -3. & 0 & 2. & 1. \\ 25. & 1. & 0 & 0 & 0 \\ 1. & 0 & 0 & 0 & 0 \\ 1. & 0 & 0 & 0 & 0 \\ 20. & 1. & 0 & 0 & 0 \\ 4. & 1. & 0 & 0 & 0 \end{bmatrix}$$

GAIN (NBLOCK)

Gain is a vector containing the gain constants of the blocks.

For the above example:

$$\text{GAIN} = [5 \quad 1 \quad K_1 \quad K_2 \quad 3 \quad 1]$$

ITHINY (< NBLOCK)

In constructing the system matrices, CLASS constructs the output equation

$$y = Hx + Fu$$

where y is an NBLOCK vector. That is, the output of each block is defined as a component of the y vector. Ordinarily, the user will want to study only a few of these outputs. The ITHINY vector "thins" out the y vector. For the above example if;

$$\text{ITHINY} = [1 \quad 3 \quad 4]$$

then $y^t = (z_1, z_3, z_4)^t$

The second option of defining a block diagram will now be described. This option allows the user to pick standard form transfer functions given in Table VI to describe the block diagram. Only one data card per block is required to describe the system. The option is chosen by setting NIT = 1 on the first card of STEP 2. The data required to specify each block are NUM, TYPE, CONNEC, MOD, PARAM where NUM is the block number.

TYPE is the type of transfer function from Table VI.

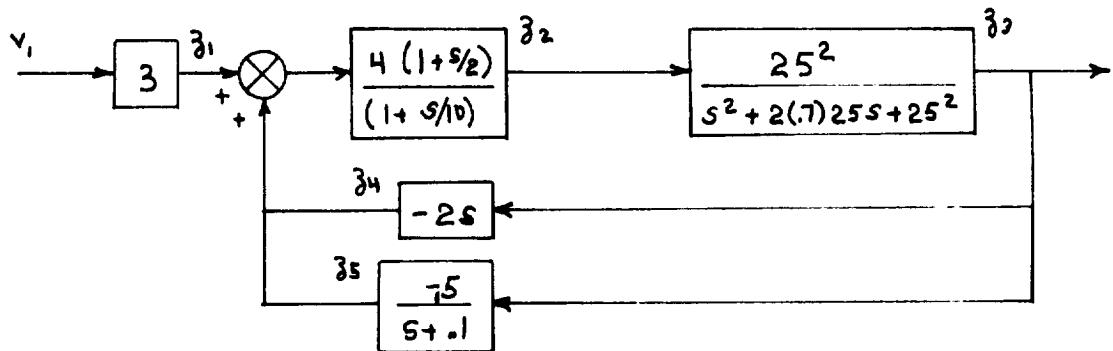
CONNEC specifies the connections between blocks and the external inputs. (This is the last four columns of GRAPH.)

MOD specifies whether the transfer function is an s , ζ , or w -transform.

PARAM gives the parameters defining the block as indicated in Table VI.

Following the cards giving the above data for each block, one card for ITWINY is read.

example: set up the input data required to describe the system



The data required are

5	1									
1	1	0	0	0	1	0	3.			
2	5	1	4	5	0	0	4.	10.	2.	
3	8	2	0	0	0	0	1.	25.	.7	
4	2	3	0	0	0	0	-2.			
5	4	3	0	0	0	0	-5.	.1		
1	2	3	4	5						

CLASS constructs a state space representation of each individual block using the phase variable canonical form. For example, the first block of the previous example:

$$G(s) = \frac{5(s^2 + 3s - 1)}{s^4 + 2s^3 - 3s + 10}$$

is modeled as

$$\dot{x} = Ax + Bu$$

$$y = Hx + Fu$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -10 & 3 & 0 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H = \begin{bmatrix} -5 & 15 & 5 & 0 \end{bmatrix} \quad F = \begin{bmatrix} 0 \end{bmatrix}$$

CLASS constructs the state space representation of the entire block diagram in the following manner:

1. Write the output vector of the collection of blocks as

$$C_3 = Hx + Fu$$

$$z_0 = C^{-1}Hx + C^{-1}Fu$$

2. Write the state equation describing the internal structure of each block (in phase variables) ignoring all connections between blocks and defining the "internal" input

vector, \underline{u}

$$\dot{x} = Ax + Bu$$

3. Define the connection matrix, G , as

$$u = Gz$$

where

$$G(i, \text{GRAPH}(i,j)) = 1 \quad \text{if } \text{GRAPH}(i,j) \neq 0 \quad j=2,3,4 \\ = 0 \quad \text{otherwise}$$

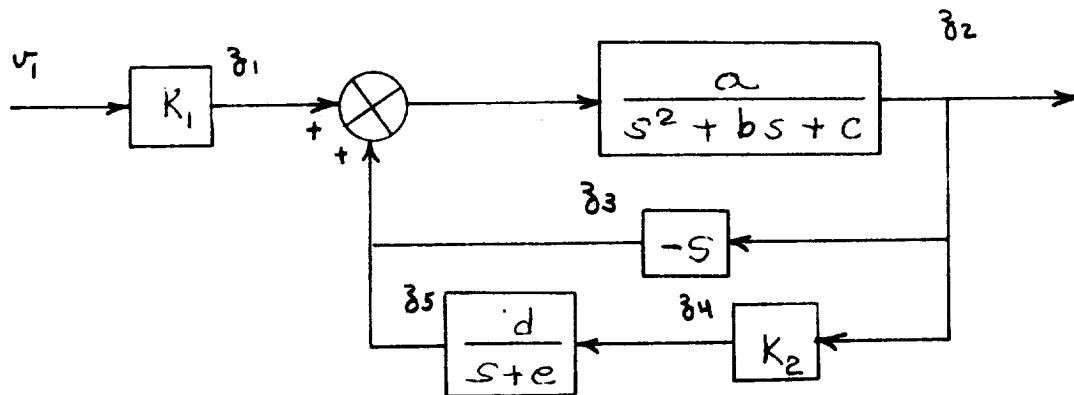
4. Couple the blocks together as

$$\dot{x} = (A + BG\bar{C}'H)x + (BG\bar{C}'F)v$$

$$y = (\bar{C}'H)x + (\bar{C}'F)v$$

The process of constructing the state variable representation is illustrated in the following example.

example: The operations described above will be performed for the system in the diagram.



$$\text{GRAPH} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & -3 & 5 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 4 & 2 & 0 & 0 & 0 \\ 5 & 4 & 0 & 0 & 0 \end{bmatrix}$$

1. Write the output equation.

$$C \vec{y} = Hx + Fu$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -K_2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{y}_1 \\ \vec{y}_2 \\ \vec{y}_3 \\ \vec{y}_4 \\ \vec{y}_5 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -a & 0 \\ 0 & 0 & 0 \\ 0 & 0 & d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} K_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} u_1 \end{bmatrix}$$

$$\vec{y} = C^{-1}Hx + C^{-1}Fu$$

$$\vec{y} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -a & 0 \\ aK_2 & 0 & 0 \\ 0 & 0 & d \end{bmatrix} x + \begin{bmatrix} K_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

2. Write the state equation of the uncoupled blocks.

$$\dot{x} = Ax + Bu$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -c & -b & 0 \\ 0 & 0 & -e \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix}$$

3. Write the connection matrix.

$$u = G \gamma$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \end{bmatrix}$$

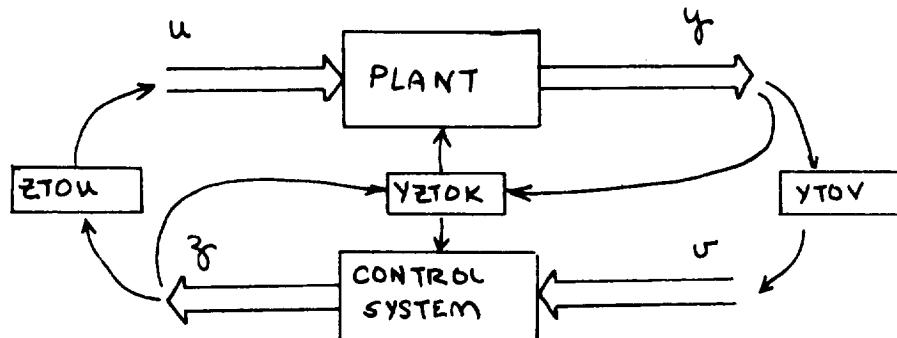
4. Couple the blocks together.

$$\dot{x} = (A + BG C^{-1} H) x + (BG C^{-1} F) v$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ -c & -(a+b) & d \\ aK_2 & 0 & -e \end{bmatrix} x + \begin{bmatrix} 0 \\ K_1 \\ 0 \end{bmatrix} v$$

MIXED INPUT OPTION (CONTINUOUS SYSTEMS)

Mixed systems are characterized by having a portion of the system described by a set of differential equations and the remainder of the system described in block diagram form. The MIXED input option allows these systems to be analyzed without requiring the user to convert the system into a common representation. Typically the plant equations of motion are given and the control system is represented in block diagram form. The MIXED option constructs the state space representation of the system in a two-step process, the first step defining the plant and the second step augmenting the system with the control system block diagram data.



The input-output pairs for the plant and control system are designated (u, y) and (v, z) , respectively. The state of the plant is x_1 , and the state of the control system is x_2 . The blocks YTOV, ZTOU, and YZTOK in the above diagram are additional connection data required by CLASS. They are used to connect the plant and controller. YZTOK defines a

feedback gain matrix to allow a conventional root locus to be performed while connections specified by YTOV and ZTOU are incorporated directly into the total system A matrix.

Table V indicates the input data required for the MIXED option.

STEP 1 The plant is modeled as an open-loop system regardless of the value of SYSTEM. (In this case, SYSTEM refers to the total, augmented system).

$$\begin{aligned} C_1 \dot{x}_1 &= A_1 \dot{x}_1 + B_1 u_1 \\ y_1 &= H_1 x_1 + G_1 \dot{x}_1 + F u_1 \end{aligned}$$

The quantities NX, NY, NU in the namelist must refer to this step. Additional dimensions required in step 2 are added by the program.

STEP 2 The control system is modeled by CLASS in the same manner described in the previous section. The required data are:

NBLOCK, GRAPH, BLOCK, NUMER, DENOM, GAIN,
ITHINY, ITHINU, NYTOV, NZTOU, NYZTOK, YTOV,
ZTOU, YZTOK.

The first seven quantities were defined in the last section while the remaining quantities specify how the system is connected.

ITHINY - Integer vector numbering, in sequence, those components of the augmented output vector $(y^T, z^T)^T$ to be saved for analysis.

IITHINU - Integer vector numbering, in sequence those components of the augmented input vector $(\mathbf{u}^T, \mathbf{v}^T)^T$ to be saved for analysis.

NYTOV - Number of connections from \mathbf{y} to \mathbf{v} .

NZTOU - " " " \mathbf{z} to \mathbf{u} .

NYZTOK - Number of feedback paths defined (with root locus option, the first connection specifies K_1 and the second connection specifies K_3 .).

YTOV - Integer matrix of dimension (NYTOV) x 2. Each row of the matrix describes a connection from \mathbf{y} to \mathbf{v} . The first element of a row specifies the component of \mathbf{y} to be connected to the element of \mathbf{v} specified by the second element.

ZTOU - Integer matrix of dimension (NZTOU) x 2 specifying connections between \mathbf{z} and \mathbf{u} in the same fashion as YTOV.

YZTOK - Integer matrix of dimension (NYZTOK) x 2 specifying a feedback gain matrix. The first number of a row specifies which element of the augmented output $(\mathbf{y}^T, \mathbf{z}^T)^T$ is to be fed back to the element of the augmented input $(\mathbf{u}^T, \mathbf{v}^T)^T$ specified by the second number of the row.

Note that numbers in YTOV and ZTOU refer to indexing in the individual vectors $\mathbf{u}, \mathbf{v}, \mathbf{y}, \mathbf{z}$ while numbers in YZTOK refer to indexing in the augmented vectors $(\mathbf{u}^T, \mathbf{v}^T)^T$ and $(\mathbf{y}^T, \mathbf{z}^T)^T$. In all cases, the indexing is specified

before any thinning of the output or input has occurred.

The CLASS subroutine constructs the state space representation of the control system with the above data. The system is:

$$\dot{x}_2 = A_2 x_2 + B_2 u$$

$$y = H_2 x_2 + F_2 u$$

The total system, at this point, is in the form of uncoupled diagonal blocks:

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & | & 0 \\ \vdots & | & \vdots \\ 0 & | & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 & | & 0 \\ \vdots & | & \vdots \\ 0 & | & B_2 \end{bmatrix} \begin{bmatrix} u \\ \vdots \\ v \end{bmatrix} \quad (7a)$$

$$\begin{bmatrix} y \\ \vdots \\ z \end{bmatrix} = \begin{bmatrix} H_1 & | & 0 \\ \vdots & | & \vdots \\ 0 & | & H_2 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_2 \end{bmatrix} + \begin{bmatrix} F_1 & | & 0 \\ \vdots & | & \vdots \\ 0 & | & F_2 \end{bmatrix} \begin{bmatrix} u \\ \vdots \\ v \end{bmatrix} \quad (7b)$$

where the dimensions are

$$x_1 - Nx$$

$$x_2 - Nx_1$$

$$y - Ny$$

$$z - Ny_1$$

$$u - Nu$$

$$v - Nu_1$$

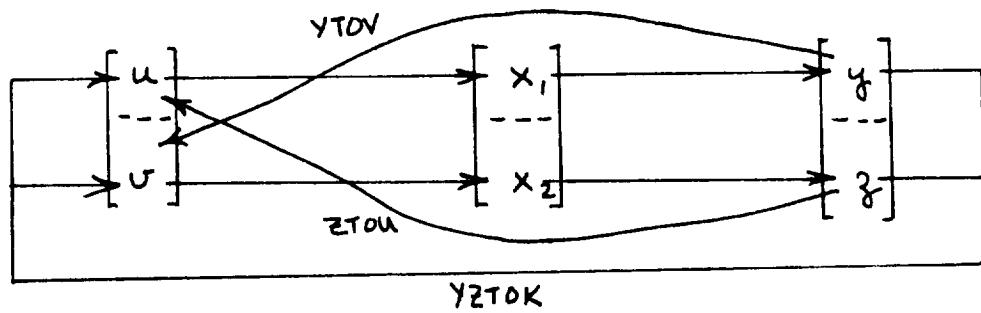
$$\begin{bmatrix} x_1 \\ \dots \\ x_2 \end{bmatrix} - NxT = Nx + NxI$$

$$\begin{bmatrix} y \\ \dots \\ \delta \end{bmatrix} - NyT = Ny + NyI$$

$$\begin{bmatrix} u \\ \dots \\ v \end{bmatrix} - NuT = Nu + NuI$$

The last three vectors above are the augmented state, augmented output, and augmented input vectors, respectively.

The system is now coupled together using YTOV, ZTOU, and YZTOK. The connections between the augmented input, state, and output vectors may be diagramed as follows.



$$\begin{bmatrix} u \\ \dots \\ v \end{bmatrix} = \begin{bmatrix} 0 & I & R_1 \\ - & - & - \\ R_2 & I & 0 \end{bmatrix} \begin{bmatrix} y \\ \dots \\ \delta \end{bmatrix} + S \begin{bmatrix} u_{com} \\ \dots \\ v_{com} \end{bmatrix} \quad (8)$$

where subscript "com" designates an external "command" input.

The dimensions of R_1 , R_2 , and S are

$$R_1 = \text{NU} \times NY1$$

$$R_2 = \text{NU1} \times NY$$

$S = \text{NUT} \times (\# \text{ of elements in ITHINU})$

R_1 , R_2 , and S are constructed according to the rules:

$$r_{ij} = \begin{cases} 1.0 & \text{if } ZTON(k,1) = j \text{ and } ZTON(k,2) = i ; k=1,2,\dots,NZTON \\ 0 & \text{otherwise} \end{cases}$$

$$r_{2ij} = \begin{cases} 1.0 & \text{if } YTOV(k,1) = j \text{ and } YTOV(k,2) = i ; k = 1, 2, \dots, NYTOV \\ 0 & \text{otherwise} \end{cases}$$

$$S_{i,j} = \begin{cases} 1.0 & \text{if } i = \text{ITHINU}(j) ; j=1,2,\dots, \\ 0 & \text{otherwise} \end{cases}$$

The submatrices R_1 and R_2 will be used to couple A_1 and A_2 while S is used to thin out the augmented input vector of unnecessary inputs. Unwanted outputs are thinned out at the very end by simply deleting rows of the final H and F output matrices (ITHINY).

The six partitioned matrices in (7) and (8) are defined to be A, B, H, F, R, and S and the system of equations is

reduced to the basic system (1) with the substitutions

$$A \leftarrow A + BR(I - FR)^{-1}H$$

$$B \leftarrow BR(I - FR)^{-1}FS + BS$$

$$H \leftarrow (I - FR)^{-1}H$$

$$F \leftarrow (I - FR)^{-1}FS$$

Now YZTOK is utilized to construct a feedback gain matrix if a root locus is desired. The feedback control law is

$$u = K_1 x + K_2 \dot{x}$$

The feedback gain matrices, K_1 and K_2 , are constructed by the rule:

let YZTOK (1, 1) = i

YZTOK (1, 2) = j

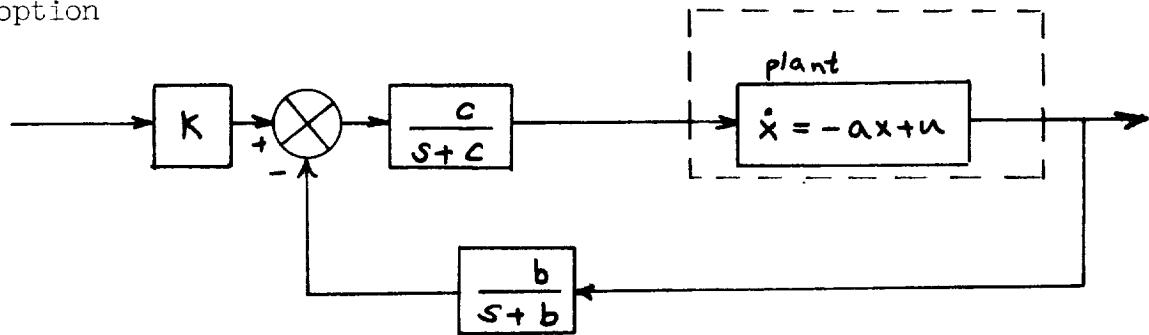
then the jth rows of K_1 and K_2 are copied from the ith row of the output matrices H and F. Thus feedback is defined from the ith output to the jth input.

If SYSTEM = 3 (root locus) a second row of YZTOK (if any) would be used to construct a second feedback variable into K_3 and K_4 , which would define the second feedback of a two-dimensional root locus.

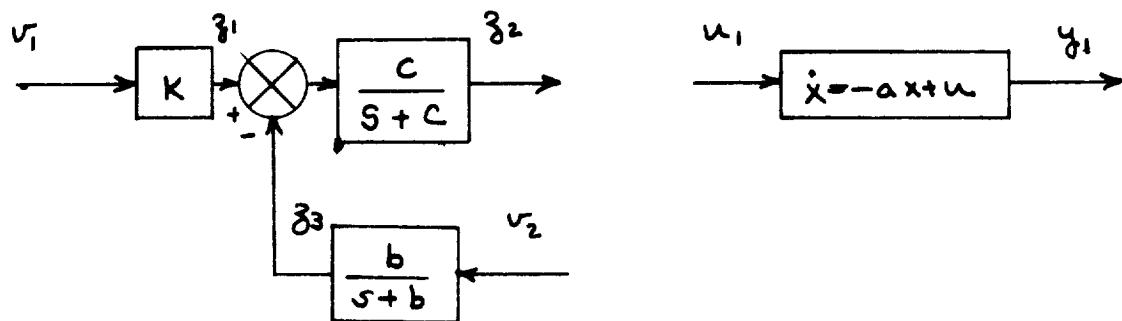
If a root locus is desired in which feedback is defined to the jth input, then the jth input cannot be thinned out of the system (i.e., j must appear in ITHINU).

If SYSTEM = 2 (closed loop) any number of feedback paths may be defined in YZTOK.

example: Set up the system indicated below using the MIXED option



The plant is described by a first order differential equation (STEP 1) and the control system is described by its block diagram representation. Redrawing the diagram:



$$\text{GRAPH} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & -3 & 0 & 0 \\ 3 & 0 & 0 & 0 & 2 \end{bmatrix}$$

a. If a closed-loop analysis is desired

$$YTOV = [1 \ 2]$$

$$ZTOU = [2 \ 1]$$

will connect the system correctly.

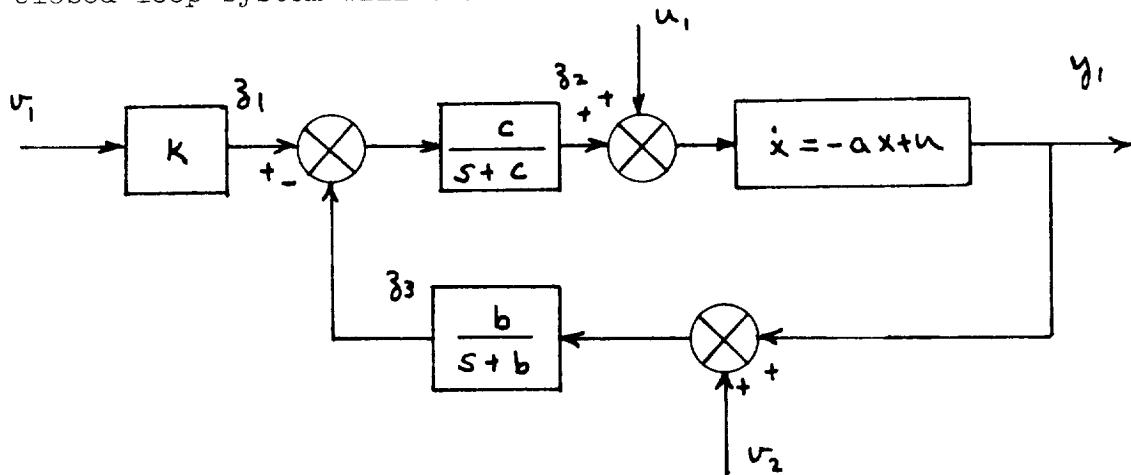
b. If a root locus is desired

$$YTOV = [1 \ 2]$$

$$YZTOK = [3 \ 1]$$

will provide the required feedback matrix (alternatively $ZTOU = [2 \ 1]$, $YZTOK = [1 \ 3]$ will also give the desired locus, defining the feedback at a different point of the system).

Returning to the closed-loop system of a., the resulting closed-loop system will be:



with

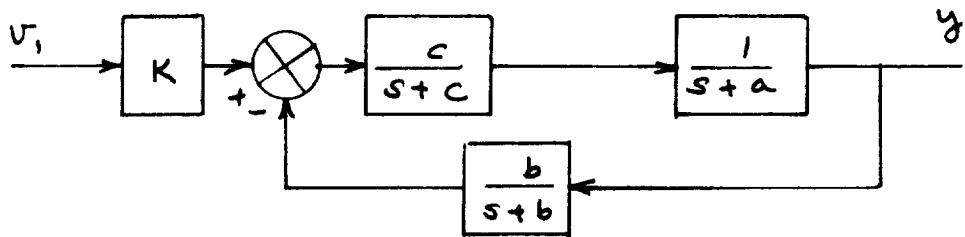
$$u = \begin{bmatrix} u_1 \\ v_1 \\ v_2 \end{bmatrix} \quad y^T = [y_1 \ z_1 \ z_2 \ z_3]$$

Notice that the connections defined by YTOV and ZTOU create summing junctions. Following the connection of the system the input label (e.g., v_2) refers to the "external input" to the summing junction and not to the "error signal" (output of the summing junction). In many cases these external inputs are not of interest and will be thinned out with ITHINU. Similar thinning will usually be done on the output vector. For instance

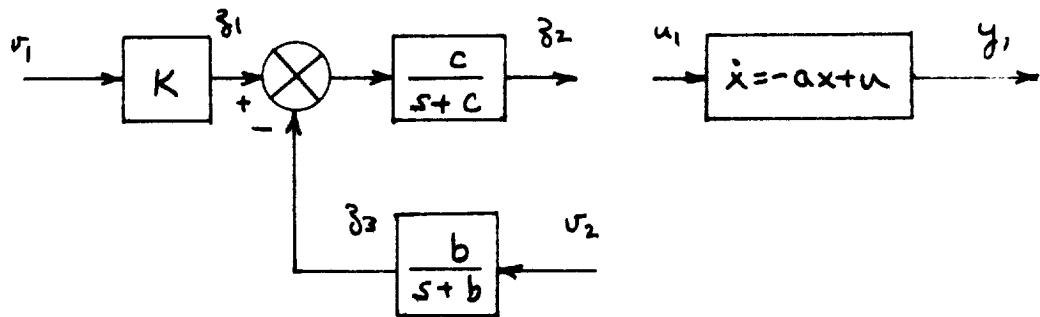
$$\text{ITHINU} = [2]$$

$$\text{ITHINY} = [1]$$

results in the system



example: set up the system of the previous example to do a root locus

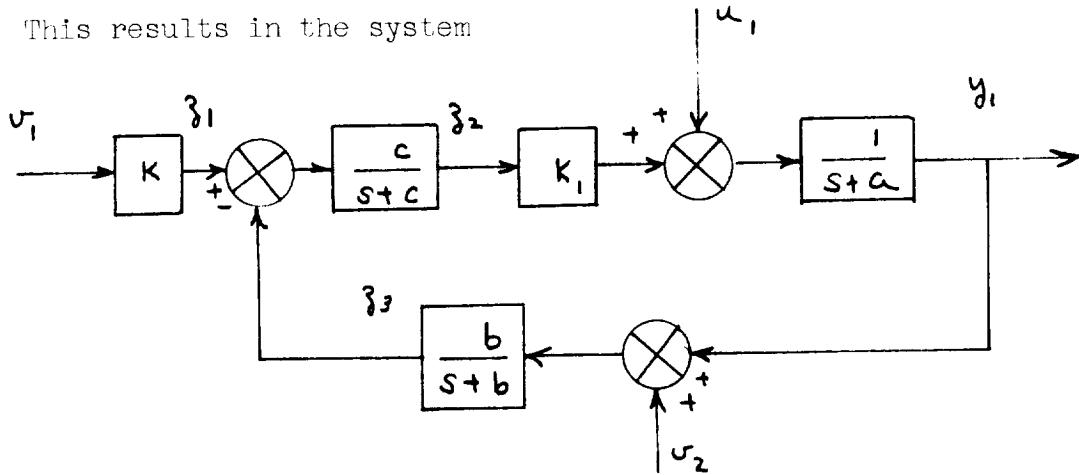


The above system can be put into the standard root locus form by defining

$$YTOV = [1 \ 2]$$

$$YZTOK = [3 \ 1]$$

This results in the system

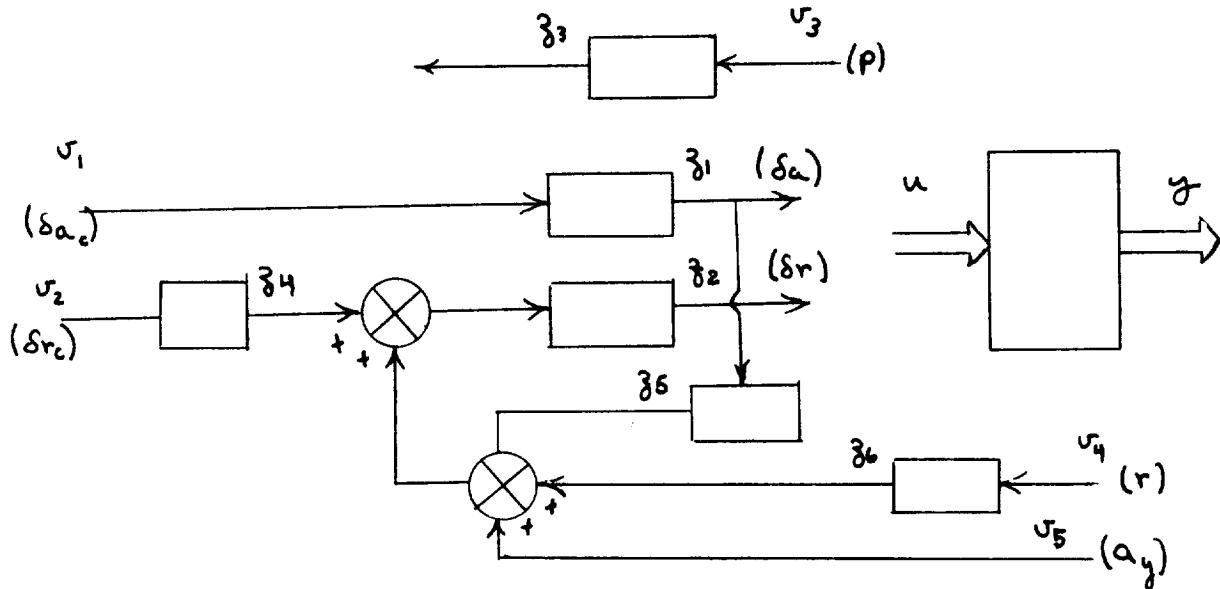


where K_1 is the root locus gain. For this system u_1 and u_2 may be thinned out but u_1 cannot be thinned out since a feedback variable has been defined to it. No output equation is defined for a root locus case, but the open-loop zeroes of ζ_2 due to an input at u_1 will be determined.

example: Suppose that STEP 1 of the MIXED option has defined a model of the lateral-directional dynamics of an airplane (A_L, B_L, H_L, F_L) with the plant input and output vectors:

$$u = \begin{bmatrix} \delta a \\ \delta r \end{bmatrix} \quad y = \begin{bmatrix} p \\ r \\ \beta \\ \phi \\ a_y \end{bmatrix}$$

It is desired to add a control system with actuator dynamics, roll rate, yaw rate, and sideforce feedbacks, and an aileron-to-rudder interconnect. Finally, a root locus of the roll rate feedback is desired with the yaw rate and sideforce feedback loops closed. The block diagram is:



(The internal dynamics of the blocks are not considered in this example since we are concerned with the method of connecting the system together.) The augmented input and output vectors are

$$u = \begin{bmatrix} \delta_a \\ \delta_r \\ \underline{\delta_a} \\ \underline{\delta_r} \\ \delta_{a_c} \\ \delta_{r_c} \\ p \\ r \\ a_y \end{bmatrix}$$

$$y = \begin{bmatrix} p \\ r \\ \rho \\ \phi \\ -\dot{a}_y \\ \delta_a \\ \delta_r \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{bmatrix}$$

The pertinent input data are

$$\text{GRAPH} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 2 & 4 & 5 & 6 & 5 \\ 3 & 0 & 0 & 0 & 3 \\ 4 & 0 & 0 & 0 & 2 \\ 5 & 1 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$\text{YTOV} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 5 & 5 \end{bmatrix}$$

$$\text{ZTOU} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$$

$$\text{YZTOK} = \begin{bmatrix} 8 & 3 \end{bmatrix}$$

The loading options which have been described above are summarized in Table V.

1. If READ = 1,2,3 and MIXED = 0, use STEP 1 of Table V.
The required matrices must be defined via LOAD, MATRIX, or CHANGE.
2. If READ = 4 and MIXED = 0, use only STEP 2 of Table V.
The required matrices will be constructed in CLASS from the block diagram data (GRAPH, etc.).
3. If MIXED = 1, the open-loop plant is defined by STEP 1 of Table V and the control system block diagram added as STEP 2 of Table V. In STEP 1, define the open-loop

plant as if SYSTEM = 1 regardless of the actual value of SYSTEM. SYSTEM, in this case, refers to the augmented system.

DISCRETE SYSTEM MODELS

If a system is known in a completely discrete form, then the system can be described by a combination of the vector difference equations,

$$x_{n+1} = Ax_n + Bu_n \quad (10a)$$

$$u_n = K_1 x_n + Du_{com,n} \quad (10b)$$

$$y_n = Hx_n + Fu_n \quad (10c)$$

The matrices G, C, K₂, and K₄ are not defined for discrete systems. This model is algebraically equivalent to the continuous system models (3), (4), and (6) and allows open-loop, closed-loop, and root locus models to be defined. The same computer algorithms which were used to generate eigenvalues for continuous systems can then be used for the discrete system. The resulting transfer functions are Z-transform transfer functions. The vector output sequence, y_n , corresponds to the continuous system transient response and may be generated directly from equations (10). The name-list parameters are defined in the same fashion as for continuous systems and all of the analysis options are available. To specify completely discrete system analysis, set DIGITL = 2 and DELT = T where T is the sample period. The completely discrete system models are given in Table IIIb. Frequency

responses are generated in the w -plane by means of the bilinear transformation

$$z = \frac{1+w}{1-w}$$

SAMPLED-DATA SYSTEM MODELS

Sampled-data systems are composed of a continuous dynamical subsystem and a discrete subsystem. The continuous subsystem is called the plant and the discrete subsystem is called the digital controller. Usually, but not always, the digital controller will be a dynamical system. To analyze a sampled-data system, the continuous plant must be discretized so that the two subsystems have a common representation.

Careful attention must be given to the structure of the system in this discretizing process and the interconnection of the two subsystems. The MIXED option assumes a central role in the sampled-data analysis with the YTOV, ZTOU, and YZTOU options used to define the connections between the subsystem.

The sampled-data system is block diagrammed in figure 5 in which $D(z)$ is the digital controller described by the triple (u_n^d, x_n^d, y_n^d) and $G(s)$ is the continuous plant described by the triple (u^c, x^c, y^c) . The plant and digital controller dynamics are given by

$$\dot{x}^c = A_c x^c + B_c u^c \quad (11a)$$

$$y^c = H_c x^c + F_c u^c \quad (11b)$$

$$\dot{x}_{n+1}^d = A_d x_n^d + B_d u_n^d \quad (12a)$$

$$y_n^d = H_d x_n^d + F_d u_n^d \quad (12b)$$

The dimension of x^c is NXC and the dimension of u^c is NUC. (The $G\dot{x}$ term in the output equation for y^c is allowed and has already been eliminated in equation (11)b. These two systems are combined in the partitioned matrix form,

$$\begin{bmatrix} \dot{x}^c \\ x_{n+1}^d \end{bmatrix} = \begin{bmatrix} A_c & 0 \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x^c \\ x_n^d \end{bmatrix} + \begin{bmatrix} B_c & 0 \\ 0 & B_d \end{bmatrix} \begin{bmatrix} u^c \\ u_n^d \end{bmatrix} \quad (13a)$$

$$\begin{bmatrix} y^c \\ y_n^d \end{bmatrix} = \begin{bmatrix} H_c & 0 \\ 0 & H_d \end{bmatrix} \begin{bmatrix} x^c \\ x_n^d \end{bmatrix} + \begin{bmatrix} F_c & 0 \\ 0 & F_d \end{bmatrix} \begin{bmatrix} u^c \\ u_n^d \end{bmatrix} \quad (13b)$$

Thus the order of the augmented state and output vectors is

- (a) plant states (outputs)
- (b) digital controller states (outputs)

CONTROL discretizes the upper left hand (NXC) x (NXC) sub-matrix of the augmented system A matrix.

The plant is generally assumed to be an open-loop system but analog feedback loops may be defined within the plant. Any such analog feedback must be defined explicitly in the plant A matrix, in YTOV, or ZTOU. Analog actuator dynamics and sensor dynamics are included in the plant. Actuators and sensors may be modeled in block diagram form using the MIXED option. The digital controller will usually be comprised

of a summing junction and digital filters. External inputs to the plant and digital controller may be defined. External inputs to the digital controller are considered to be sequences of numbers, u_{ext}^d , separated in time by the sample period, T . External inputs to the plant, u_{ext}^c , are considered as sampled continuous inputs. The inputs to the plant are comprised of the outputs of the digital controller and the sampled external inputs. These inputs may be applied to the plant as

- (a) outputs of samplers (pulse trains)
- (b) outputs of zero-order-hold elements,

The sampled-data block diagram of figure 5 is capable of representing a wide range of sampled-data systems. Figure 6 shows several of the possible configurations. Figure 6(a) shows an open-loop plant with a sampled (pulse train) input, figure 6(b) shows a closed-loop plant with a sampled error signal, figure 6(c) shows a closed-loop plant with digital compensation in the feedback path, and figure 6(d) shows a closed-loop plant with digital compensation in the forward path. Questions arising in the analysis of these systems involves the stability of the closed-loop system, the system transient response, and the synthesis of (digital) control systems. The CONTROL program allows these questions to be studied using the ζ -plane root locus, the w -plane frequency response (both standard and modified ζ -transforms), and the system transient response.

In order to perform this analysis, it is necessary to discretize the plant to obtain a discrete system model of the entire sampled-data system.

The form of the input function determines the proper discretization of the plant. CONTROL treats the first ZOH elements of u^c as inputs from zero-order-hold elements and the remaining (NUC - ZOH) elements of u^c as sampled inputs. Thus the ordering of the augmented input vector is

- (a) zero-order-hold inputs to plant
- (b) sampled inputs to plant
- (c) discrete inputs to digital controller

CONTROL discretizes the upper left hand (NXC) x (NUC) submatrix of the augmented system B matrix. The first ZOH columns of the submatrix are discretized to account for the zero-order-hold effect and the remaining (NUC-ZOH) columns are discretized to account for the sampling effect.

The system which results from the discretization of the plant is block diagramed in figure 7. All of the sequences x_n^d , x_n^c , y_n^c , etc., are defined at the same instants of time

$$t = nT ; \quad n=0,1,2,\dots$$

However, since the system actually contains a continuous subsystem which has a continuous state trajectory it is necessary to define the exact meaning of sequences such as y_n^c . For instance, y_n^c has the two interpretations

$$y_n^{c^+} = \lim_{\epsilon_2 \rightarrow 0} y^c(nT + \epsilon_2) \quad \epsilon_2 > 0 \quad (14a)$$

$$y_n^{c^-} = y^c(nT - \epsilon_1) \quad \epsilon_1 > 0 \quad (14b)$$

Which of these two definitions is used has an important bearing on the resulting discretized system. The CONTROL program assumes

$$x_n^c \equiv \lim_{\epsilon \rightarrow 0} x^c(nT - \epsilon) \quad \epsilon > 0$$

$$u_n^c \equiv u^c(nT)$$

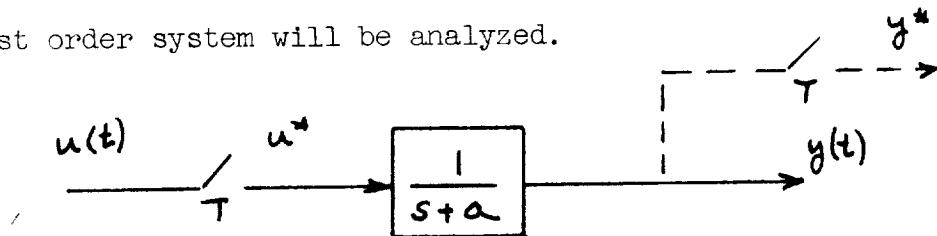
The interpretation of these definitions is that; the state, x_n^c , is defined at time $t = nT$ prior to the application of the input, u_n^c . The updated input, u_n^c , is applied to the system at the sampling instant, $t = nT$. Thus all events are timed with respect to u_n^c as occurring before or after the application of u_n^c .

Figure 8(a) shows the time sequence of events in the complete sampled-data system. At $t = nT - \epsilon_1$, the plant is sampled to provide $y_n^{c^-}$ and $u_{ext,n}^d$ is input to the digital controller. Following the computational delay, ϵ_1 , the digital controller output, y_n^d , is updated (using $y_n^{c^-}$ and $u_{ext,n}^d$), the plant input, u_n^c , is defined (using y_n^d and $u_{ext,n}^c$) and applied to the plant. At $t = nT + \epsilon_2$ the plant output is $y_n^{c^+}$. The time delay ϵ_2 may be regarded as

an infinitesimal (i.e., the limit goes to zero in eq. (14(a)) since the plant responds instantaneously to u_n^* . The time delay ϵ_1 cannot be regarded as an infinitesimal since the digital controller requires a finite time to compute the update. y_n^d . However, if $\epsilon_1 \ll T$ it is customary to assume $\epsilon_1 \rightarrow 0$.

Figure 8(b) shows the idealized time sequence for the sampled-data system. In this idealized model, events occur only at the sample instants and they occur instantaneously.

To illustrate the discretizing process an open-loop first order system will be analyzed.



The state equation of the plant driven by the input train is

$$\dot{x} = -\alpha x + u$$

$$u = u^* = u(t) \delta(t - nT) \quad n=0,1,2, \dots$$

The solution of the state equation for $0 \leq t < T$ is

$$\begin{aligned} x(t) &= \phi(t)x_0 + \int_0^t \phi(t-\tau) u^*(\tau) d\tau \\ &= e^{-\alpha t} x_0 + \int_0^t e^{-\alpha(t-\tau)} \delta(\tau) u(\tau) d\tau \end{aligned}$$

$$= e^{-at} x_0 + e^{-at} u_0$$

and in general

$$x_{n+1} \equiv x[(n+1)T] = e^{-aT} x_n + e^{-aT} u_n \quad (15a)$$

The output equation depends upon the choice of output; y_n^- or y_n^+ .
At $t = nT - \epsilon$ ($\epsilon \neq 0$) $u^* = 0$ and

$$y_n^- = x_n \quad (15b)$$

At $t = nT + \epsilon$ the impulse at $t = nT$ will have changed
the state x_n to

$$\begin{aligned} x_n^+ &= x(nT + \epsilon) = \lim_{\epsilon \rightarrow 0^+} \left[\phi(\epsilon) x_n + \int_{nT}^{nT+\epsilon} \phi(nT + \epsilon - \tau) u^*(\tau) d\tau \right] \\ &= \lim_{\epsilon \rightarrow 0^+} \left[\phi(\epsilon) x_n + \phi(\epsilon) u_n \right] \end{aligned}$$

$$x_n^+ = x_n + u_n$$

since $\phi(0) = 1$ and $u^*(\tau) = u_n \delta(\tau - nT)$

The output y_n^+ is then

$$y_n^+ = x_n + u_n \quad (15c)$$

since $u(nT + \epsilon) = 0$.

The discretized system is given by equations (15a) and (15b)

if y_n^- is chosen as the output, or by (15a) and (15c) if y_n^+
is chosen as the output. The corresponding z-transfer func-
tions are

$$G(z) = \frac{e^{-\alpha T}}{z - e^{-\alpha T}} \quad \text{if } y_n \equiv y_n^- \quad (16a)$$

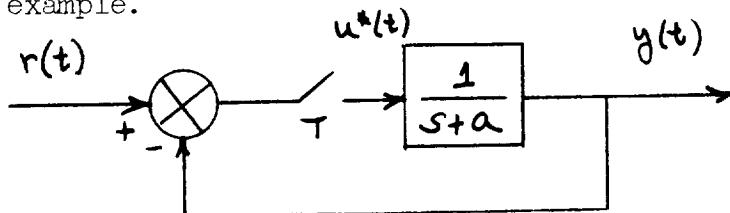
$$G(z) = \frac{z}{z - e^{-\alpha T}} \quad \text{if } y_n \equiv y_n^+ \quad (16b)$$

An interesting result of this example is that (16b) is the standard z -transfer function (pulse transfer function) of $G(s) = \frac{1}{s+a}$. Thus to generate standard z -transfer functions for the plant (11) the following convention should be used:

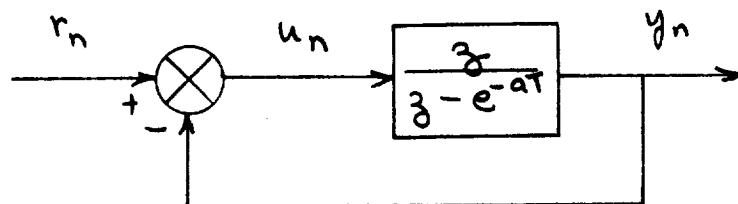
$$y_n \equiv y_n^+ = y(nT + \epsilon) \quad (17a)$$

$$x_n \equiv x_n^- = x(nT - \epsilon) \quad (17b)$$

Although this convention will generate correct pulse transfer functions, it is not the convention required for analyzing closed-loop sampled-data systems. This is evident from the following example.



This system is replaced by the equivalent discrete system:



whose state equation is (assuming $y_n = y_n^+$)

$$x_{n+1} = e^{-\alpha T} x_n + e^{-\alpha T} u_n$$

$$y_n = x_n + u_n$$

The control law is $u_n = r_n - y_n = r_n - x_n - u_n$

This control law is nonphysical since it requires the digital controller to compute, at $t = nT - \epsilon$, a control law requiring data not available until $t = nT$ (see fig. 8). Thus for a closed-loop sampled data system the convention $y_n \equiv y_n^+$ can lead to a nonphysical realization of the system. The correct convention for closed-loop sampled-data systems is

$$y_n \equiv y_n^- = y(nT - \epsilon) \quad \epsilon > 0$$

since this equation represents the measurements (observations) available to the digital controller at the time at which it must compute the control law.

Table VII lists the correctly discretized plant models for the various input and output definitions. Table VII(a) gives the models for the output convention $y_n \equiv y_n^-$ and Table VII(b) gives the models for the output convention $y_n \equiv y_n^+$. In Table VII(a) $F = 0$. This is required for the pulsed input since a continuous system with $F \neq 0$ cannot be driven by a pulse train. The ZOH input system (Table VII(a)) cannot have $F \neq 0$ since this would make y_n dependent on u_{n-1} and the analysis options of CONTROL are not applicable to such a system.

The only differences in Table VII(a) and (b) are in the direct feedforward term of the output equations. It is interesting to note that these terms in Table VII(b) ($H_B u_n$ for pulsed inputs and $F_u n$ for ZOH inputs) will be zero if

- all input-output transfer paths have at least 2 more poles than zeroes for the pulsed input system
- all input-output transfer paths have at least 1 more pole than zeroes for the ZOH system.

Inputs from digital controllers to continuous mechanical plants are almost always applied through zero-order-hold elements whose outputs drive actuators. If actuator dynamics are included in the plant then there will always be at least 1 more pole than zeroes for all transfer paths and there will be no difference in the two ZOH input cases of Table VII(a) and (b).

From Table VI the operations required to discretize the continuous plant are now apparent. CONTROL replaces the augmented system model of equations (13) with

$$\begin{bmatrix} x_{n+1}^c \\ x_{n+1}^d \end{bmatrix} = \begin{bmatrix} \phi & 0 \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x_n^c \\ x_n^d \end{bmatrix} + \underbrace{\begin{bmatrix} [\Phi B_c] & [\Phi B_d] & 0 \\ 0 & 0 & B_d \end{bmatrix}}_{\text{ZOH}} \begin{bmatrix} u_n^c \\ u_n^d \end{bmatrix} \quad (18a)$$

$$\begin{bmatrix} y_n^c \\ y_n^d \end{bmatrix} = \begin{bmatrix} H_c & 0 \\ 0 & H_d \end{bmatrix} \begin{bmatrix} x_n^c \\ x_n^d \end{bmatrix} + \underbrace{\begin{bmatrix} [F_c] & [H_c B_c] & 0 \\ 0 & 0 & F_d \end{bmatrix}}_{\text{NUC}} \begin{bmatrix} u_n^c \\ u_n^d \end{bmatrix} \quad (18b)$$

The partitioned submatrices $([\oplus B_c], [\phi B_c], [F_c], [H_c B_c])$ are interpreted as follows: The first $N \times C$ rows and $Z \times H$ columns of $\oplus B_c$ are copied into the corresponding location of the B matrix, etc. At this point the sampled-data system is completely discretized and is given by

$$x_{n+1} = Ax_n + Bu_n \quad (19a)$$

$$y_n = Hx_n + Fu_n \quad (19b)$$

where the augmented vectors and matrices are:

$$x_n = \begin{bmatrix} x_n^c \\ \vdots \\ x_n^d \end{bmatrix} \quad u_n = \begin{bmatrix} u_n^c \\ \vdots \\ u_n^d \end{bmatrix} \quad y_n = \begin{bmatrix} y_n^c \\ \vdots \\ y_n^d \end{bmatrix}$$

$$A = \begin{bmatrix} \phi(\tau) & | & 0 \\ \hline 0 & | & A_d \end{bmatrix} \quad B = \begin{bmatrix} [\oplus B_c] & | & [\phi B_c] & | & 0 \\ \hline 0 & | & 0 & | & B_d \end{bmatrix}$$

$$H = \begin{bmatrix} H_c & | & 0 \\ \hline 0 & | & H_d \end{bmatrix} \quad F = \begin{bmatrix} [F_c] & | & [H_c B_c] & | & 0 \\ \hline 0 & | & 0 & | & F_d \end{bmatrix}$$

In light of the preceding discussion concerning the possibility of defining nonphysical feedback control laws, the submatrices $[F_c]$ and $[H_c B_c]$ must be interpreted carefully. CONTROL does not use $[F_c]$ or $[H_c B_c]$ to

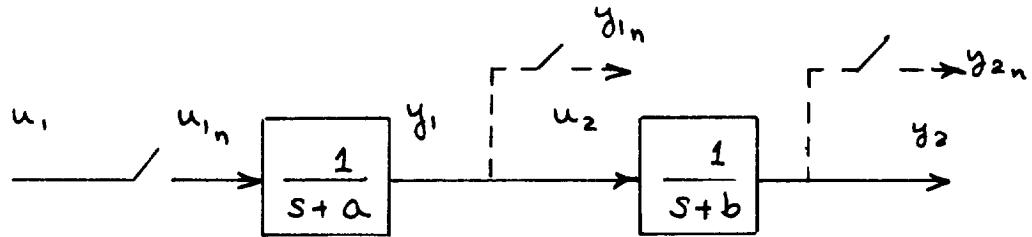
define connections from the plant to the discrete subsystem or to define sampled-data feedback laws. This amounts to using $y_n^c = y^c(nT - \epsilon)$ instead of $y_n^c = y^c(nT + \epsilon)$ for these connections.

These submatrices are used in computing \mathcal{Z} -transfer functions, frequency responses, and transient responses. Thus pulse transfer functions such as that of the above example can be computed. This completes the discussion concerning the discretization of the plant dynamics.

The sampled-data system has been reduced to a set of uncoupled matrix difference equations (19). The remaining step in the definition of the system is the method of connecting the digital controller and the discretized plant. These connections are best illustrated by the following examples. The examples illustrate the various possible combinations of continuous and discrete subsystems. Each example contains two systems. The appropriate uncoupled state and output equations are written for each example followed by the equation giving the connection between the two systems. The state equations are discretized, the connection equation incorporated at the appropriate time, and the final coupled discretized system is given.

example: Determine the properly discretized and connected equations for the following systems.

a.)



1. Original system.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -a & 0 \\ 0 & -b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

2. Connection equation.

$$u_2 = y_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

3. Coupled system.

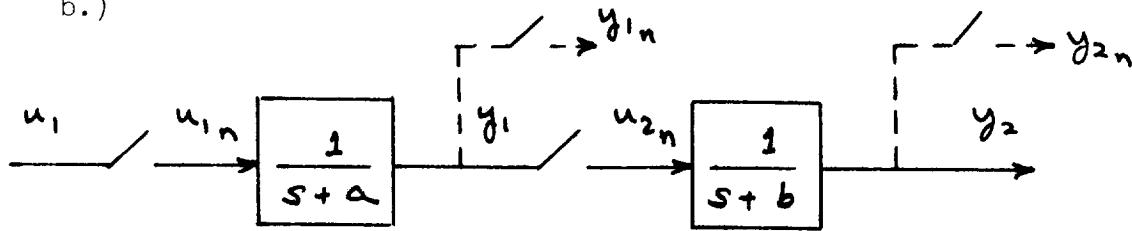
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -a & 0 \\ 1 & -b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1$$

4. Final system.

$$\begin{bmatrix} x_{1,n+1} \\ x_{2,n+1} \end{bmatrix} = \begin{bmatrix} e^{-aT} & 0 \\ \frac{e^{-aT}-e^{-bT}}{(b-a)} & e^{-bT} \end{bmatrix} \begin{bmatrix} x_{1,n} \\ x_{2,n} \end{bmatrix} + \begin{bmatrix} e^{-aT} \\ 0 \end{bmatrix} [u_{1,n}]$$

$$\begin{bmatrix} y_{1,n} \\ y_{2,n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,n} \\ x_{2,n} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [u_{1,n}]$$

b.)



1. Original system.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -a & 0 \\ 0 & -b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

2. Discretized system.

$$\begin{bmatrix} x_{1n+1} \\ x_{2n+1} \end{bmatrix} = \begin{bmatrix} e^{-aT} & 0 \\ 0 & e^{-bT} \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} + \begin{bmatrix} e^{-aT} & 0 \\ 0 & e^{-bT} \end{bmatrix} \begin{bmatrix} u_{1n} \\ u_{2n} \end{bmatrix}$$
$$\begin{bmatrix} y_{1n}^+ \\ y_{2n}^+ \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1n} \\ u_{2n} \end{bmatrix}$$

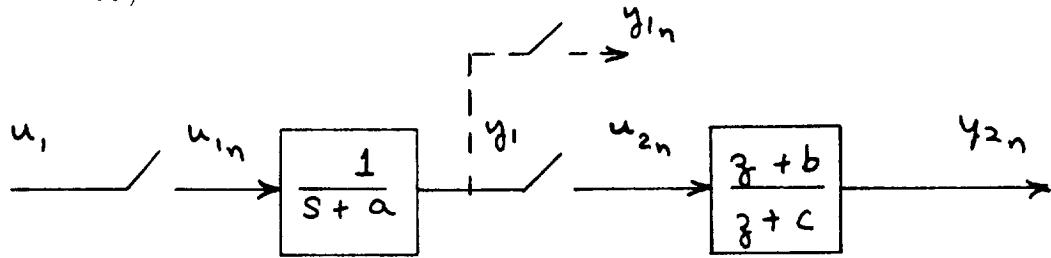
3. Connection equation.

$$u_{2n} = y_{1n}^- = [1 \quad 0] \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix}$$

4. Final system.

$$\begin{bmatrix} x_{1n+1} \\ x_{2n+1} \end{bmatrix} = \begin{bmatrix} e^{-aT} & 0 \\ 0 & e^{-bT} \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} + \begin{bmatrix} e^{-aT} \\ 0 \end{bmatrix} u_{1n}$$
$$\begin{bmatrix} y_{1n}^+ \\ y_{2n}^+ \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_{1n}$$

c.)



1. Original system.

$$\begin{bmatrix} \dot{x}_1 \\ x_{2n+1} \end{bmatrix} = \begin{bmatrix} -a & 0 \\ 0 & -c \end{bmatrix} \begin{bmatrix} x_1 \\ x_{2n} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_{2n} \end{bmatrix}$$

$$\begin{bmatrix} y_1^+ \\ y_{2n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & (b+c) \end{bmatrix} \begin{bmatrix} x_1 \\ x_{2n} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_{2n} \end{bmatrix}$$

2. Discretized system.

$$\begin{bmatrix} x_{1n+1} \\ x_{2n+1} \end{bmatrix} = \begin{bmatrix} e^{-aT} & 0 \\ 0 & -c \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} + \begin{bmatrix} e^{-aT} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1n} \\ u_{2n} \end{bmatrix}$$

3. Connection equation.

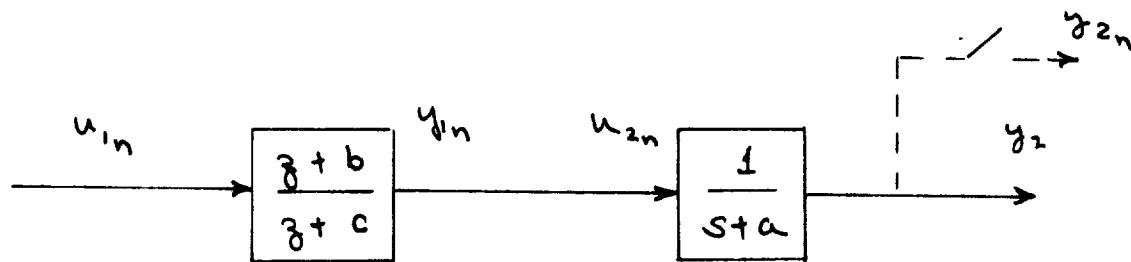
$$u_{2n} = y_{1n}^+ = [1 \quad 0] \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix}$$

4. Final system.

$$\begin{bmatrix} x_{1n+1} \\ x_{2n+1} \end{bmatrix} = \begin{bmatrix} e^{-aT} & 0 \\ 1 & -c \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} + \begin{bmatrix} e^{-aT} \\ 0 \end{bmatrix} u_{1n}$$

$$\begin{bmatrix} y_{1n}^+ \\ y_{2n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & (b+c) \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_{1n}$$

d.)



1. Original system.

$$\begin{bmatrix} \dot{x}_2 \\ x_{1n+1} \end{bmatrix} = \begin{bmatrix} -a & 0 \\ 0 & -c \end{bmatrix} \begin{bmatrix} x_2 \\ x_{1n} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ u_{1n} \end{bmatrix}$$

$$\begin{bmatrix} y_2 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & (b-c) \end{bmatrix} \begin{bmatrix} x_2 \\ x_{1n} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ u_{1n} \end{bmatrix}$$

2. Discretized system.

$$\begin{bmatrix} x_{2n+1} \\ x_{1n+1} \end{bmatrix} = \begin{bmatrix} e^{-aT} & 0 \\ 0 & -c \end{bmatrix} \begin{bmatrix} x_{2n} \\ x_{1n} \end{bmatrix} + \begin{bmatrix} e^{-aT} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{2n} \\ u_{1n} \end{bmatrix}$$

$$\begin{bmatrix} y_{2n}^+ \\ y_{1n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & (b-c) \end{bmatrix} \begin{bmatrix} x_{2n} \\ x_{1n} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{2n} \\ u_{1n} \end{bmatrix}$$

3. Connection equation.

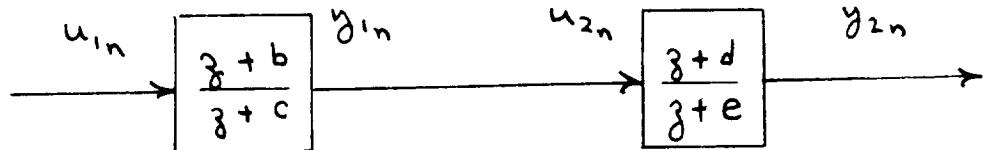
$$u_{2n} = y_{1n} = [0 \ (b-a)] \begin{bmatrix} x_{2n} \\ x_{1n} \end{bmatrix} + [0 \ 1] \begin{bmatrix} u_{2n} \\ u_{1n} \end{bmatrix}$$

4. Final system.

$$\begin{bmatrix} x_{2n+1} \\ x_{1n+1} \end{bmatrix} = \begin{bmatrix} e^{-aT} (b-c)e^{-aT} \\ 0 \ -c \end{bmatrix} \begin{bmatrix} x_{2n} \\ x_{1n} \end{bmatrix} + \begin{bmatrix} e^{-aT} \\ 1 \end{bmatrix} \begin{bmatrix} u_{1n} \end{bmatrix}$$

$$\begin{bmatrix} y_{2n}^+ \\ y_{1n} \end{bmatrix} = \begin{bmatrix} 1 & (b-c) \\ 0 & (b-c) \end{bmatrix} \begin{bmatrix} x_{2n} \\ x_{1n} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} u_{1n} \end{bmatrix}$$

e.)



1. Original system.

$$\begin{bmatrix} x_{1,n+1} \\ x_{2,n+1} \end{bmatrix} = \begin{bmatrix} -c & 0 \\ 0 & -e \end{bmatrix} \begin{bmatrix} x_{1,n} \\ x_{2,n} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1,n} \\ u_{2,n} \end{bmatrix}$$
$$\begin{bmatrix} y_{1,n} \\ y_{2,n} \end{bmatrix} = \begin{bmatrix} (b-c) & 0 \\ 0 & (d-e) \end{bmatrix} \begin{bmatrix} x_{1,n} \\ x_{2,n} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1,n} \\ u_{2,n} \end{bmatrix}$$

2. Connection equation.

$$u_{2,n} = y_{1,n} = [(b-c) \quad 0] \begin{bmatrix} x_{1,n} \\ x_{2,n} \end{bmatrix} + [1 \quad 0] \begin{bmatrix} u_{1,n} \\ u_{2,n} \end{bmatrix}$$

3. Final system.

$$\begin{bmatrix} x_{1,n+1} \\ x_{2,n+1} \end{bmatrix} = \begin{bmatrix} -c & 0 \\ b-c & -e \end{bmatrix} \begin{bmatrix} x_{1,n} \\ x_{2,n} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} u_{1,n} \\ u_{2,n} \end{bmatrix}$$

$$\begin{bmatrix} y_{1,n} \\ y_{2,n} \end{bmatrix} = \begin{bmatrix} b-c & 0 \\ b-c & d-e \end{bmatrix} \begin{bmatrix} x_{1,n} \\ x_{2,n} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} u_{1,n} \\ u_{2,n} \end{bmatrix}$$

A close study of these examples reveals that:

- i.) Connections between two continuous systems must be made before they are discretized (a).
- ii.) Connections to a continuous system from a sampler, zero-order-hold, or digital system must be made after the continuous system is discretized (b and d). Such a continuous system must have an input defined and the connection to the input is specified in a "feedback" gain matrix.
- iii.) Connections to a digital system from a continuous system may be made either before or after the system is discretized (c). The discretization has no effect on the connection equation.
- iv.) Connections between discrete systems may be made at any time since the discretization does not affect these systems (e).

SAMPLED-DATA SYSTEM ANALYSIS USING THE MIXED OPTION

The use of the MIXED option of the CONTROL program to model sampled-data systems may be described now. The MIXED option involves a two-step process in which a set of linear differential equations are augmented with a block diagram control system.

Sampled-data systems involve a continuous subsystem, the plant, and a discrete subsystem, the digital controller. The CONTROL program constructs the sampled-data system model

by defining the plant in STEP 1 and the digital controller in STEP 2 of the MIXED option. It is not required that the plant correspond completely with STEP 1 of the MIXED option. For example, the block diagram control system (STEP 2) may contain a mixture of Laplace transformed blocks (actuator and sensor dynamics) and ζ -transformed blocks (the digital filters of the digital controller). The Laplace transform blocks are thus part of the plant. Conversely, difference equations defining digital filters may be written explicitly in the A matrix in STEP 1. The required ordering of inputs, outputs, and states is given in Table VIII(a). This ordering produces a preliminary sampled-data system which is a composite of the system of equations (7) and (13). (Due to the possibility of defining part of the plant in STEP 2, the correspondence between vectors (e.g., x_i in eq. (7) and x^c in eq. (13)) is not exact.)

This preliminary system is connected together by connections specified in YTOV, ZTOU, and YZTOK. In light of the results of the examples given above, provision must be made for making some connections before the plant is discretized and the remainder after the discretization. To achieve this, the following sequence of operations is performed:

- i.) YTOV and ZTOU connections completed
- ii.) plant discretized
- iii.) YZTOK defines feedback gain matrix, K_l
- iv.) feedback gain matrix, K_l , incorporated into total

system A matrix (closed-loop analysis) or used to perform a root locus

Thus connections which must be completed before the system is discretized are defined in YTOV, ZTOU, GRAPH, or the A_1 matrix of STEP 1. Connections from discrete systems to the plant (which must be made after the plant is discretized) must be defined in YZTOK. Table VIII(b) summarized the types of connections allowed and how they must be defined. $G(s)$ and $D(z)$ represent continuous and discrete subsystems, respectively. The sampled-data system model and its construction are given in Table III(c). The step indicated in Table III(c)(4) is the critical step in constructing the sampled-data system model. This step connects the digital controller to the plant in the correct manner. If a root locus is desired, this step defines the appropriate feedback gain matrix. If a root locus is called for, then the first connection specified by YZTOK will generate the feedback gain matrix, K_1 , and the second element of YZTOK (if any) will generate a second feedback gain matrix, K_3 . The control law is then

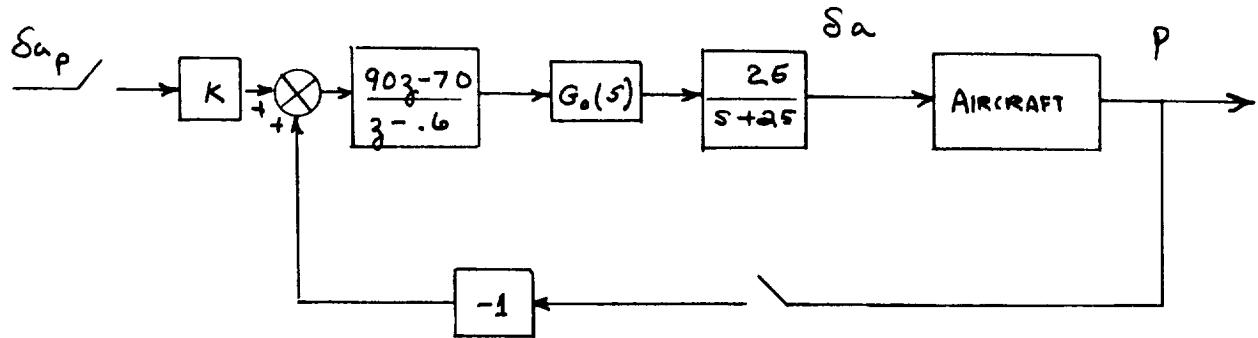
$$u_n = K_1 x_n + K_3 x_n$$

and K_1 , K_3 would generate a root locus grid.

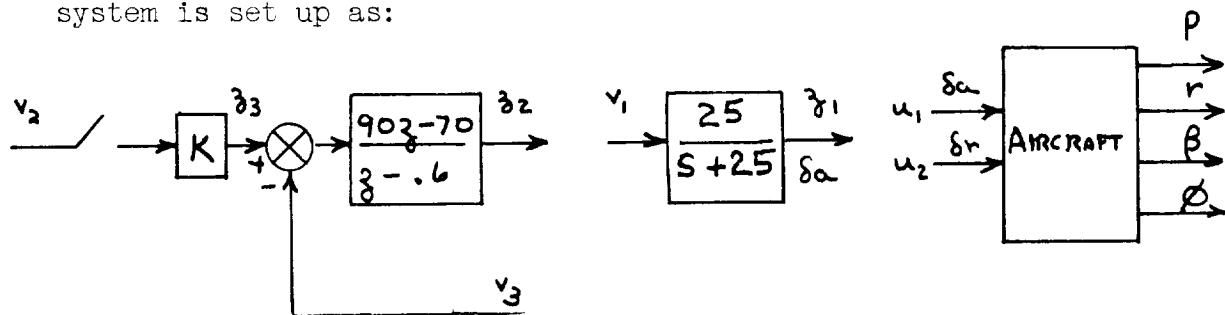
The formulation of the sampled-data system is illustrated in the following examples.

example: A lateral-directional aircraft plant with a roll rate feedback to a digital controller is to be modeled. The

block diagram of the system is:



The block diagram is comprised of a pilot input gain, a digital filter, a zero-order-hold, and the aileron actuator transfer function. The aircraft state vector and output vector is $y = x = (\rho \ r \ \beta \ \phi)^T$ and the aircraft equations of motion are input to CONTROL in STEP 1. The system is set up as:



STEP 1 defines the aircraft with $u = (\delta_a \ \delta_r)^T$. In STEP 2 the actuator block input and output must be numbered first since it is part of the plant. The appropriate data for STEP 2 is:

NBLOCK

3

GRAPH

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 2 & 3 & 0 & 0 & -3 \\ 3 & 0 & 0 & 0 & 2 \end{bmatrix}$$

BLOCK

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 2 \\ 3 & 1 & 1 \end{bmatrix}$$

NUMER

$$\begin{bmatrix} 25. \\ -70. 90. \\ 1. \end{bmatrix}$$

DENOM

$$\begin{bmatrix} 25. 1. \\ -.6 1. \\ 1. \end{bmatrix}$$

GAIN

$$[1. \quad 1. \quad K]$$

ITHINY

$$[1 \quad 2 \quad 3 \quad 4]$$

ITHINU

$$[3 \quad 4]$$

NYTOV, NZTOU,
NZTOK

$$[1 \quad 1 \quad 1]$$

$$YTOV \quad [1 \quad 3]$$

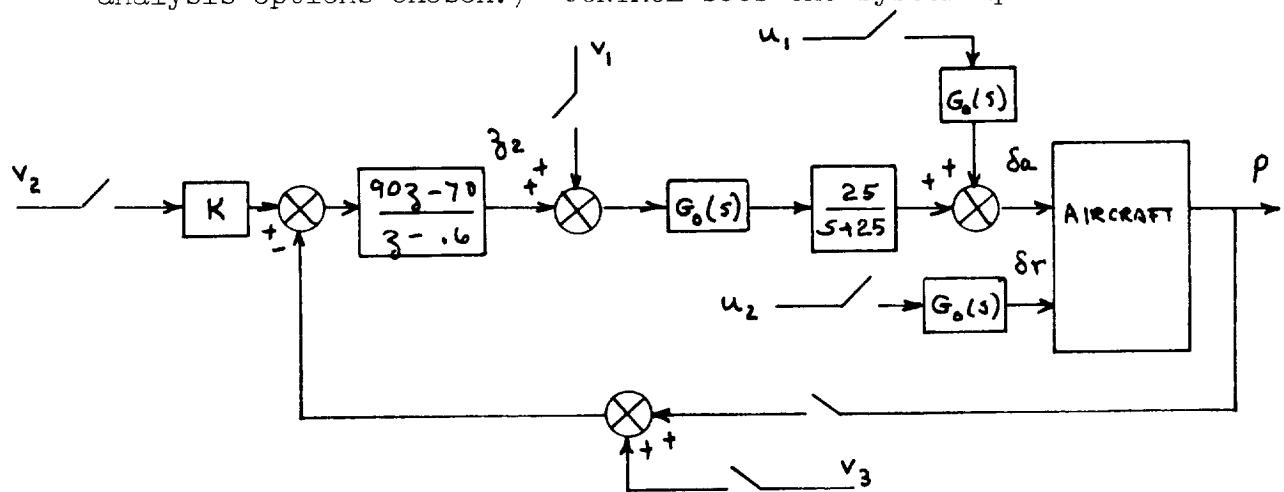
$$ZTOU \quad [1 \quad 1]$$

$$YZTOK \quad [6 \quad 3]$$

The pertinent namelist parameters are

$$\text{DIGITL} = 1, \text{NXC} = 5, \text{NUC} = 3, \text{ZOH} = 3, \dots$$

(Other parameters are required to specify STEP 1 and the analysis options chosen.) CONTROL sets the system up as



The connection from δ_2 to the summing junction before the zero-order-hold is constructed as a feedback control law

$$u = \begin{bmatrix} \delta_a \\ \delta_r \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = K_1 x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -90.0.0. & 0. & 0. & 0 & -K_0. \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \phi \\ \delta_a \end{bmatrix}$$

If SYSTEM = 2, this "feedback" connection is completed and closed-loop ζ -transfer functions, w -plane frequency responses, or transient responses may be obtained. If SYSTEM = 3, the feedback gain matrix is used to calculate the ζ -plane root locus.

The ITHINU vector in the above example specifies that only the 3rd and 4th elements of the augmented input vector will be retained. The 3rd element (v_1) must be retained if SYSTEM = 3 since a feedback is defined to this input. If v_1 is not retained the root locus could not be obtained.

COMPUTATIONAL DELAY AND MODIFIED ζ -TRANSFORMS

The digital controller requires a finite time, ϵ_1 , to compute the updated command signals to the plant (fig. 8(a)). The preceding analysis has assumed that $\epsilon_1 \ll T$. If this condition is not met the computational delay may be critical to the system stability. The modified ζ -transform allows this case to be analyzed using open-loop frequency response techniques. A linear analysis of a sampled-data system requires that all events pertinent to the system (state, input, output) be defined at one instant in time, the sampling instant. But if $\epsilon \approx T$, the digital filter computes commands based on the system output at time $nT - \epsilon_1$, while the plant state is defined only at time nT . The situation is shown in figure 9. At $t = (n+m-1)T$, the digital controller samples the plant output, $y_n^c(m)$, and inputs

u_n^d . The digital controller then computes the updated plant input u_n^c . This requires $(1-m)T$ seconds where $0 \leq m \leq 1$. If $m = 1$ there is no delay and if $m = 0$ there is a one sample period delay.

The output $y_n^c(m)$ is dependent upon the state of the plant at $t = (n-1)T$ and the form of the input u_n^c from $t = (n-1)T$ to $t = (n+m-1)T$. If the input to the plant is from a sampler then

$$\begin{aligned} y_n^c(m) &= H x_n^c[(n+m-1)T] \\ &= H \left[\phi(mT) x_{n-1}^c + \phi(mT) B u_{n-1}^c \right] \end{aligned} \quad (20a)$$

where

$$\phi(mT) = \int_0^{mT} e^{-A(mT-\tau)} d\tau$$

If the input to the plant is from a zero-order-hold then

$$y_n^c(m) = H \phi(mT) x_{n-1}^c + [H \oplus(mT) B + F] u_{n-1}^c \quad (20b)$$

where

$$\oplus(mT) = \int_0^{mT} \phi(mT-\tau) d\tau$$

In either case, $y_n^c(m)$ is described by (20a) or (20b) as a linear combination of x_{n-1}^c and u_{n-1}^c . Modified z-transform analysis is performed by idealizing the time sequence model of figure 9 and considering $y_n^c(m)$ and the digital controller update that it generates to occur instantaneously

with the plant update at $t = nT$. The resulting system model for sampled inputs is

$$x_{n+1}^c = \phi(\tau) x_n^c + \phi(\tau) B u_n^c \quad (21a)$$

$$y_n^c(m) = H \phi(m\tau) x_{n-1}^c + H \phi(m\tau) u_{n-1}^c \quad (21b)$$

The ζ -transform of (21) yields the transfer function matrix

$G(\zeta, m)$ where

$$\begin{aligned} Y_m(\zeta) &\equiv G(\zeta, m) U(\zeta) \\ &= \frac{1}{\zeta} \left\{ H \phi(m\tau) [(\zeta I - \phi(\tau))^{-1} \phi(\tau) B + B] \right\} U(\zeta) \end{aligned}$$

Thus the model of equations (21) will generate modified ζ -transfer functions with the identifications

$$\begin{aligned} A &\leftarrow \phi(\tau) \\ B &\leftarrow \phi(\tau) B \\ H &\leftarrow H \phi(m\tau) \\ F &\leftarrow H \phi(m\tau) B \end{aligned}$$

A similar transfer matrix may be derived for zero-order-hold inputs.

The resulting ζ -transfer functions which the CONTROL program generates must be multiplied by ζ^m to account for the sample period delay in the output equation (21b).

example: find $G(\zeta, m)$ for $G(s) = \frac{1}{s+a}$

The plant equations are

$$\dot{x} = -\alpha x + u$$

$$y = x$$

then $\phi(t) = e^{-\alpha t}$, $\phi(mT) = e^{-\alpha mT}$ and

$$x_{n+1} = e^{-\alpha T} x_n + e^{-\alpha T} u_n$$

$$y_n(m) = e^{-\alpha mT} x_{n-1} + e^{-\alpha mT} u_{n-1}$$

$$Y_m(z) = \frac{1}{z} \left\{ e^{-\alpha mT} \frac{e^{-\alpha T}}{z - e^{-\alpha T}} + e^{-\alpha mT} \right\} U(z)$$

$$= \frac{1}{z} \frac{\frac{e^{-\alpha mT}}{z - e^{-\alpha T}} U(z)}{z - e^{-\alpha T}}$$

$$\text{or } G(z, m) = \frac{e^{-\alpha mT}}{z - e^{-\alpha T}}$$

To obtain modified z -transforms of open-loop sampled-data systems the parameter m must be specified in the namelist.

If $m = 1$, the standard z -transform will result if $y_n^c = y_n^{c-}$.

Numerical errors limit m to $m > .2$. All other namelist

parameters required for sampled-data system analysis are unchanged.

Only open-loop analysis can be accomplished with $m \neq 0$. Digital filters may be cascaded with the "open loop"

plant using the MIXED option as described above. If the

digital filter output drives the plant, recall that a quasi

"closed-loop" system is set up with YZTOK defining a "feed-

back" law and SYSTEM must be set to 2. Table VII(c) lists

the modified z -transform models.

METHOD OF ANALYSIS

The methods of analysis used by the CONTROL program are summarized in this section.

Frequency Response and Power Spectra

The basic system equations ((1.) or (2.)) may be transformed to yield

$$\rho X(\rho) = AX(\rho) + BU_{com}(\rho) \quad (22a)$$

$$Y(\rho) = HX(\rho) + FU_{com}(\rho) \quad (22b)$$

where $\rho = s$ or z depending on the type of system. Equation (22) is rewritten to display the system transfer matrix,

$$Y(\rho) = G(\rho)U_{com}(\rho) \quad (23)$$

where

$$G(\rho) = H(\rho I - A)^{-1}B + F$$

The transfer matrix, $G(\rho)$, is an $NY \times NU$ matrix of transfer functions. If transfer functions and/or frequency responses are requested, CONTROL computes these $(NY) \cdot (NU)$ functions. Frequency responses are generated at discrete frequencies in the following manner:

1. Continuous systems - $G(\rho) = G(s)$. The frequency response of the i^{th} output due to the j^{th} input is generated by setting $s = j\omega$ in $g_{ij}(s)$ and allowing ω to take on discrete values as specified in Appendix 2.
2. Discrete and sampled-data systems - $G(\rho) = G(z)$
Frequency responses of z -transformed functions are

accomplished either in the w -plane or the ζ -plane under the control of FRPS. If FRPS = 1, $G(\zeta)$ is transformed to a w -plane transfer matrix, $G(w)$, by the transformation

$$\zeta = \frac{1+w}{1-w}$$

The frequency response is accomplished by the substitution $w = j\gamma$ at discrete points along the positive imaginary axis in the w -plane.

If FRPS = -1, the frequency response is accomplished by the substitution $\zeta = \cos \omega T + j \sin \omega T$ at discrete points along the upper unit semicircle in the ζ -plane.

The advantage of the w -plane frequency response is that asymptotic Bode plot methods may be used. (This is due to the frequency response being a polynomial function of the frequency.)

3. Continuous power spectra - $G(p) = G(s)$

Power spectra are computed for continuous systems from the relation

$$S_{y_i}(\omega) = |g_{ij}(s)|^2 S_{u_j}(\omega)$$

where $S_{y_i}(\omega)$ and $S_{u_j}(\omega)$ are the power spectra of the i^{th} output and j^{th} input, respectively, and $g_{ij}(s)$ is the corresponding transfer function. CONTROL assumes that

$$S_{u_j}(\omega) = 1$$

i.e., the input is unity variance white noise. Thus, to compute power spectra, a "shaping filter" will usually be added to the system dynamics and driven by the white noise input. The output of the filter then drives the system with the desired spectral content.

Eigenvalues

CONTROL uses the QR algorithm to determine the system eigenvalues. HESSEN transforms the matrix to upper Hessenburg form. QREIG then determines the eigenvalues via calls to QRT. The subroutines QREIG, HESSEN, and QRT are contained in the IBM Share Program No. 3006.01 written by P. Imiad Fawzi and J. E. VanNess, Northwestern University. The QR algorithm is discussed in reference 1.

Transfer Function Numerators

CONTROL determines the transfer function numerators as the eigenvalues of a matrix derived from the A, B, H, and F matrices. Details may be found in reference 2.

Transient Responses and Discretization of Sampled-Data Systems

To compute transient responses of continuous systems and to discretize sampled-data systems, the transition matrix and its integral are required. They are computed in the EAT subroutine by summing the partial series

$$\Phi(t) = e^{At} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \dots + \frac{1}{n!}A^n t^n$$
$$\Theta(t) = \int_0^t e^{A(t-\tau)} d\tau = It + \frac{1}{2!}At^2 + \frac{1}{3!}A^2t^3 + \dots + \frac{1}{n!}A^{(n-1)}t^n$$

The series are terminated when the last terms in both series cause changes to each element of both series less than 10^{-3} times the respective element or when the series has not converged in 24 terms.

In the computation of ϕ for sampled-data systems, it is common to have eigenvalues whose magnitudes are comparable to the half-sample frequency resulting in slow convergence of the series. In this case, the user does not have the flexibility of using a smaller integration step size since the sample period is fixed. To help alleviate this problem, CONTROL computes $\phi(\tau/8)$ and $\oplus(\tau/8)$ and then finds

$\phi(\tau)$ and $\oplus(\tau)$ as

$$\phi(\tau) = [\phi(\tau/8)]^8$$

$$\oplus(\tau) = \oplus(\tau/8) \left[\sum_{i=0}^7 \phi(i\tau/8) \right]$$

Details may be found in reference 3.

Transient responses for continuous systems are calculated using $\phi(\tau)$ and $\oplus(\tau)$ as

$$x[(n+1)\tau] = \phi(\tau)x(n\tau) + \oplus(\tau)Bu(n\tau)$$

$$y(n\tau) = Hx(n\tau) + Fu(n\tau)$$

where $u(n\tau)$ is defined in the INPUT subroutine. The input, $u(n\tau)$, is held constant between sample periods.

Transient responses for discrete and sampled-data systems are computed in a similar manner from the difference equations

given in Table VII.

For sampled-data systems, the MULTRT option allows the user to compute the intersample response of the system. The system is then described by:

$$\begin{aligned}x_{n+1} &= Ax_n + Bu_n \\y_n &= Hx_n + Fu_n \\u_n &= K_1 x_n + Du_{\text{com}_n}\end{aligned}$$

where A and B are obtained by discretizing the plant for the time period, T/MULTRT and u_n is updated every T seconds. That is, the plant is discretized as though the sample period was T/MULTRT but u_n is held constant over T seconds. Thus, MULTRT intersample transient response points will be computed. Only transient responses are allowed with this option.

Digital Filtering

In synthesizing sampled-data systems, much time and effort can go into the computation of digital filter coefficients to give desired filtering to a signal. The CONTROL program allows the user to choose from a table of standard filters (Table VI) the filtering action he desires. The filter may be specified in the s , ζ , or w -plane. The transformation of s - and w -plane filters to ζ -plane filters can be carried out automatically by the program, allowing the user to draw upon experience in analog filtering techniques. The transformation of a w -plane filter to a ζ -plane filter is accomplished by replacing w by

$$w = \frac{z-1}{z+1}$$

The transformation of an s -plane filter to a w -plane filter is

$$u = \tanh\left(\frac{\alpha T}{2}\right)$$

where α is the s -plane first order pole or zero and u is the corresponding w -plane pole or zero. For complex poles and zeroes

$$\left(1 + \frac{2g}{\omega} s + \frac{s^2}{\omega^2}\right)$$

the transformation is

$$\left(1 + \frac{2g_w}{\omega_w} w + \frac{w^2}{\omega_w^2}\right)$$

where

$$\omega_w = \sqrt{u^2 + v^2}$$

$$g_w = -u/\omega_w$$

$$u = \frac{\sinh(\alpha T)}{\cosh(\alpha T) + \cos(\beta T)}$$

$$v = \frac{\sin(\alpha T)}{\cosh(\alpha T) + \cos(\beta T)}$$

with

$$\omega^2 = \alpha^2 + \beta^2$$

$$\alpha = -g_w$$

Digital filters derived from this two-step process (called the bilinear transformation with frequency prewarping)

maintain a close resemblance of the original s -plane filters over a wide range of the half-sample frequency. References 4 and 5 give discussions of digital filters.

Model Following - If MODEL = 1, CONTROL computes frequency responses appropriate to the evaluation of model following systems. Let y_1 be the model output, y_2 be the model follower output, and let the model have inputs u_1 , and u_2 . The CONTROL program will compute the frequency responses (with FRPS set appropriately)

$$1 \quad \frac{y_1}{u_1}(j\omega)$$

$$2 \quad \frac{y_2}{u_1}(j\omega) ; \quad \frac{y_2/u_1}{y_1/u_1}(j\omega)$$

$$3 \quad \frac{y_1}{u_2}(j\omega)$$

$$4 \quad \frac{y_2}{u_2}(j\omega) ; \quad \frac{y_2/u_2}{y_1/u_2}(j\omega)$$

Thus, with MODEL set, CONTROL will provide the frequency responses

$$\frac{y_{2i+1}/u_2}{y_{2i}/u_2}(j\omega) \quad \left\{ \begin{array}{l} i = 1, 2, \dots, Ny/2 \\ k = 1, 2, \dots, Nu \end{array} \right.$$

in addition to the standard frequency responses.

Similarity Transformation - Numerical problems in calculating eigenvalues can sometimes be traced to the eigenvalues of a matrix being small compared to its norm. In this event, diagonal similarity transformations may aid in eigenvalue computation. For the autonomous system

$$\dot{x} = Ax \quad (24)$$

a similarity transformation is defined as the change of variables

$$z = Px$$

for P nonsingular. The system (24) becomes

$$\dot{z} = P\dot{x} = PAP^{-1}z = A'z$$

with $A' = PAP^{-1}$. The matrix A' has the same eigenvalues as A and if P is chosen properly, A' will have a smaller norm than A .

The CONTROL program includes an option (NSCALE) which the user may select if numerical problems are suspected in eigenvalue computations. In order to maintain the same input-output relationship, the program performs the following operations when the A matrix is scaled.

$$\begin{aligned} PB &\rightarrow B \\ HP^{-1} &\rightarrow H \\ K1P^{-1} &\rightarrow K1 \\ K3P^{-1} &\rightarrow K3 \end{aligned}$$

The scaling technique is described in references 6 and 7.

PROGRAM SIZE AND TIMING

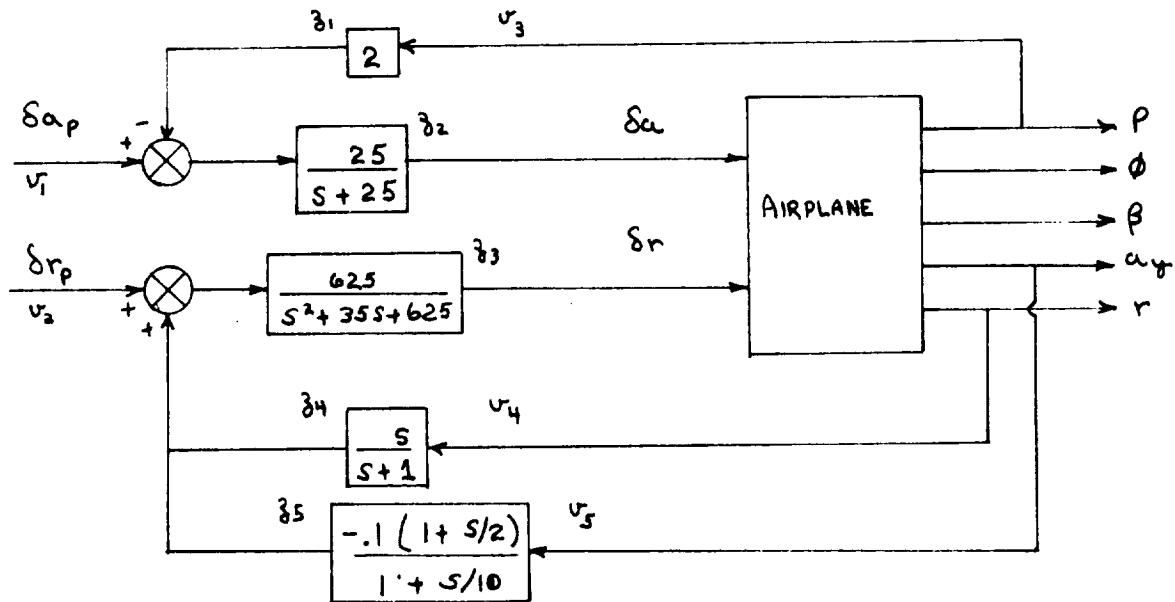
The CONTROL program requires 131 K₈ words of computer memory. It has operated on a CDC CYBER 70 computer utilizing a segmented structure which reduced the memory requirement to 72 K₈ words. Included in these size requirements is memory allocated to data matrices for maximum state, output, and input

vector dimensions of 15, 15, and 10, respectively. The data matrix storage requirements are a function of the system dimensions.

The example program of the next section was executed on the CDC CYBER 70 and required 17 sec. of CPU execution time to generate the problem setup and calculate ten transfer functions, ten frequency responses, and a transient response.

EXAMPLE PROBLEM

An example problem is given to illustrate the problem formulation, data deck, and output listing. The input and output listings are given in Appendix 3. The problem involves a lateral-directional airplane model with a control system consisting of aileron and rudder actuators, a roll rate feedback to δ_a , and yaw rate and side force feedback to δ_r through a washout and lead-lag filter, respectively. The system is shown in the block diagram.



The airplane equations of motion are

$$\begin{bmatrix} \dot{\rho} \\ \dot{r} \\ \dot{\beta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} L_p & L_r & L_\beta & 0 \\ N_p & N_r & N_\beta & 0 \\ d_t & -1 & Y_\beta & g/v \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ r \\ \beta \\ \phi \end{bmatrix} + \begin{bmatrix} L_{\delta a} & L_{\delta r} \\ N_{\delta a} & N_{\delta r} \\ 0 & Y_{\delta r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta a \\ \delta r \end{bmatrix}$$

$$\begin{bmatrix} p \\ r \\ \beta \\ \dot{\phi} \\ a_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -Y_g d_t & V/g & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & V/g & 0 \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{r} \\ \dot{\beta} \\ \dot{\phi} \end{bmatrix}$$

the control system is described using the MIXED option and frequency responses and transient responses are obtained for roll rate and yaw rate.

CONCLUDING REMARKS

A FORTRAN digital computer program for the analysis of linear continuous and sampled-data systems has been described. The program features a flexible input format allowing the program user to define systems in a variety of representations. All systems are analyzed using state variable techniques. Analysis options of the program are: transfer functions, frequency responses, power spectra, root loci, root contours, and transient responses.

Dryden Flight Research Center
National Aeronautics and Space Administration
Edwards, Calif.. January 12, 1976

REFERENCES

1. Parlett, B. N.: The LU and QR Algorithms. Mathematical Methods for Digital Computers, Volume II, Anthony Ralston and Herbert S. Wilf, eds., John Wiley & Sons, Inc., c.1967, pp. 116-130.
2. Sandberg, Irwin W.; and So, Hing C.: A Two-Sets-of-Eigenvalues Approach to the Computer Analysis of Linear Systems. IEEE Trans. Circuit Theory, vol. CT-16, no. 4, Nov. 1969, pp. 509-517.
3. Johnson, J. C.; and Phillips, C. L.: An Algorithm for the Computation of the Integral of the State Transition Matrix. IEEE Trans. Automatic Control, vol. AC-16, no. 2, Apr. 1971, pp. 204-205.
4. Kaiser, J. F.: Digital Filters. System Analysis by Digital Computer, Franklin F. Kuo and James F. Kaiser, eds., John Wiley & Sons, Inc., c.1966, pp. 218-285.
5. IEEE Trans. Audio and Electroacoustics. Vol. AU-16, no. 3, Sept. 1968.
6. Osborne, E. E.: On Pre-Conditioning of Matrices. J. Assn. for Computing Machinery, vol. 7, c.1960, pp. 338-345.
7. Harary, Frank: A Graph Theoretic Method for the Complete Reduction of a Matrix With a View Toward Finding Its Eigenvalues. J. Math. Phys., vol. 38, 1959, pp. 104-111.

APPENDIX 1
FORTRAN LISTING OF CONTROL

A brief description of the various subroutines of CONTROL follows:

CONTROL - is the MAIN subroutine as described on page 5.

ADD - performs matrix addition.

ASCALE - scales the A matrix with a diagonal similarity transformation.

CARD - is described on page 4.

CHANGE - is described on page 6.

CLASS - is described on page 4.

CNTRLR - serves as the executive routine for CONTROL.

CPMT - computes complex roots.

EAT - computes the transition matrix and its integral.

EIGEN - is described on page 4

FRQRSP - is described on page 4.

HESSEN - transforms a matrix to the upper Hessenburg form.

INPUTV - is the INPUT subroutine described on page 6.

INVR - determines the inverse of a matrix.

LOAD - is described on page 4.

LOAD1 - is used in conjunction with LOAD to load the system matrices.

MAKE - makes two matrices equivalent.

MATRIX - is described on page 4.

MULT - performs matrix multiplication.

NMRATR - is described on page 4.

PSP - is described on page 4.

OREIG - determines the system eigenvalues.

QRT - is used in conjunction with OREIG.

RDISC - reads input matrices from the disc storage units.

RDISCl - is used in conjunction with RDISC.

REDUCE - determines the irreducible submatrices of a matrix (used with ASCALE).

ROOT - is described on page 4.

SETUP - is described on page 4.

SPIT - outputs matrices on the printer.

SPIT1 - is used in conjunction with SPIT.

THIST - is described on page 5.
(starts at label SPIT1 16)

SWZ - transforms s-and w-plane filters to z-plane filters.

TANG - computes complex arc tangents.

WDISC - writes input matrices on to disc storage.

WDISCl - is used in conjunction with WDISC.

ZOT - initializes the system matrices to zero.

ZOT1 - is used in conjunction with ZOT.

ZTOW - converts z-plane transfer functions to w-plane transfer functions.

COP0 - plots data on CALCOMP plotter.

READ0 - plots zeros for a root locus.

CSTAR - contains envelope curves for the C* options.

SUBSCL - computes the scaling factor for root locus plots.

PROGRAM CONTROL 13/74 OPT=1

FTN 4.2+75060 01/09/76 13.59.41.

```

      PROGRAM CONTROL (INPUT=6,OUTPUT=6,TAPE1=INPUT,TAPE 3=OUTPUT,TAPE 5
      1=6,TAPE6=65,TAPE7=67,TAPE8=68,TAPE9=69)
      REAL K1,K2,K3,K4
      DIMENSION W1(15,15),W2(15,15),W3(15,15),ROOTR(15),ROOTI(15),
      1ROTPI(15),ROTI(15),SAV1(200),SAV2(200),U(15),V(15)
      DIMENSION A(15,15),B(15,10),C(15,15),H(15,15),G(15,1),F(15,10),
      1K(10,15),K2(10,15),K3(10,15),K4(10,15),D(10,10)
      COMMON/SURNAM/TSURNAME
      DATA TSURNAME/0/
      10      C
      C      CONTROL MAIN PROGRAM. ABSOLUTE DIMENSIONS FOR ALL ARRAYS ARE
      C      DECLARED HERE AND PASSED THROUGHOUT THE PROGRAM IN THE COMMON
      C      BLOCK. TO CHANGE DIMENSIONS OF ARRAYS REQUIRES CHANGES ONLY
      C      IN THIS ROUTINE. NX MUST BE SET EQUAL TO 'NX=1' WHERE NX IS THE
      C      MAXIMUM DIMENSION OF THE STATE VECTOR.
      C
      C      IF (TSURNAME .NE. 2) WRITE (3,9901
      990 FORMAT(1X, *MAIN*)
      MX=15
      20      MY=10
      MU=10
      MS=200
      CALL  CNTRLR (A,B,C,H,G,F,K1,K2,K3,K4,O,W1,W2,W3,ROOTR,ROOTI,
      1ROTPI,ROTI,SAV1,SAV2,U,V,
      2MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      STOP
      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE ADD 77/74 CPT=1 FTN 4, 2478060 01/09/75 13:59:47.
 SUBROUTINE ADD (X,A,Y,B,C,N,M,
 1 PX, PY, MU, MS, MAT1, MAT2, MAT3, MAT4, MAT5, MAT6)
 DIMENSION A(MAT1,MAT2),B(MAT3,MAT4),C(MAT5,MAT6)
 COMMON/SUBRPT/ ICUB3AM
 IF(ICUB3AM.GE.2) WRITE(3,9901)
 990 FORMAT(1X,*ADD*)
 DO 10 I=1,N
 DO 10 J=1,M
 C(I,J)=X*A(I,J)+Y*B(I,J)
 10 CONTINUE
 RETURN
 END

	1.00	2
	A00	3
	A00	4
	A00	5
	A00	6
	A00	7
	A00	8
	A00	9
	A00	10
	A00	11
	A00	12
	A00	13

ORIGINAL PAGE IS
OF POOR QUALITY

```

      SUBROUTINE ASCALE (N,J,M,MM,P,A,B,C,DIFAR,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      C
      C THIS SUBROUTINE SCALES THE MATRIX A (OF DIMENSION N) WITH
      C A DIAGONAL SIMILARITY TRANSFORMATION. SUBROUTINE REDUCE IS
      C CALLED TO DETERMINE THE LFR/EUCILINE SUBMATRICES OF A.
      C
      C J= NUMBER OF IRREDUCIBLE SUBMATRICES
      C MM(I)= DIMENSION OF THE ITH SUMMATRIX
      C M = MATRIX WHOSE ITH ROW CONTAINS THE ORIGINAL ROW AND COLUMN
      C NUMBERS OF THE ITH SUMMATRIX
      C TRANSFORMATION MATRIX
      C P = VECTOR CONTAINING THE DIAGONAL ELEMENTS OF THE
      C
      C THE ROUTINE PRODUCES THE SCALED A MATRIX
      COMMON/COND/READ,SYSTEM,OUTPUT,NX,NY,NU,NXD,NUC,N1,N2,DIGITL,
     1CCNTUR,NUMERS,FRPS,TRESP,MODEL,NSCALE,SAV,CMAT,NK2,IFLAG,
     2IGC,FORM,IPT,READ3,MIXED,MULTRI,SCAPLT,ZOM,KOUNT
      INTEGER READ,SYSTEM,OUTPUT,FCRP,CCNTUR,SAV,CMAT,READ3, FRPS,TRESP
      20 INTEGER DIGITL,SCAPLT,70H
      DIMENSION DIBAR (MX)
      DIMENSION A(MX,MX),B(MX,MX),C(MX,MX)
      DIMENSION M(10,20),MM(20),P(20)
      CCMCN/SUBWRT/ ISUBNAM
      25 IF (ISURNAM.GE.2) WRITE (3,*90)
      990 FORMAT(1X,*SCALE*)
      NSCALE=0
      ITEST=0
      CALL PDUCE (N,J,MM,M,A,B,C,ITEST,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF (ITEST.EQ.0) GO TO 110
      DO 111 I=1,N
      P(I)=1.0
      111 CONTINUE
      WRITE (3,112)
      112 FORMAT (/* SIMILARITY TRANSFORMATION MATRIX SET EQUAL TO IDENTI
     1TY*/)
      RETURN
      110 CONTINUE
      DO 75 I=1,N
      P(I)=0.0
      75 CONTINUE
      STOP=1.0
      DO 100 I=1,J
      NI=MM(I)
      IF (NI.NE.1) GO TO 70
      P(M(I,I))=1.0
      GO TO 100
      70 CONTINUE
      DO 50 II=1,NI
      DO 51 JJ=1,NI
      B(II,JJ)=A(M(I,II),M(I,JJ))
      51 CONTINUE
      DIRAP(II)=1.0
      P(M(I,II))=1.0
      50 CONTINUE
      ICTRC=0

```

ASCALE 2
ASCALE 3
ASCALE 4
ASCALE 5
ASCALE 6
ASCALE 7
ASCALE 8
ASCALE 9
ASCALE 10
ASCALE 11
ASCALE 12
ASCALE 13
ASCALE 14
ASCALE 15
ASCALE 16
ASCALE 17
ASCALE 18
ASCALE 19
ASCALE 20
ASCALE 21
ASCALE 22
ASCALE 23
ASCALE 24
ASCALE 25
ASCALE 26
ASCALE 27
ASCALE 28
ASCALE 29
ASCALE 30
ASCALE 31
ASCALE 32
ASCALE 33
ASCALE 34
ASCALE 35
ASCALE 36
ASCALE 37
ASCALE 38
ASCALE 39
ASCALE 40
ASCALE 41
ASCALE 42
ASCALE 43
ASCALE 44
ASCALE 45
ASCALE 46
ASCALE 47
ASCALE 48
ASCALE 49
ASCALE 50
ASCALE 51
ASCALE 52
ASCALE 53
ASCALE 54
ASCALE 55
ASCALE 56
ASCALE 57
ASCALE 58

SUBROUTINE ASCALE 7174 OPT=1

FTN 4.2+75060 01/09/76 13.59.45.

```

      K=0
50  DO 10 II=1,NI
      SAPI=0.0
      RAID=0.0
      DO 73 IP=1,NI
      IF ((II,IP).GT.20) GO TO 20
      RAIP=RAID+B(II,IP)**2
      SAPI=SAPI+B(IP,II)**2
10  CONTINUE
      DITTY= SORT(SAPI/RAIP)
      DIRAR(II)= SORT(DITTY)
      DIRARI=1.0/DIRAR(II)
      P(M(I,I))=P(M(I,I))*DIRAR(II)
      DO 30 L=1,NI
      R(II,L)=R(II,L)*DIRAR(II)
30  CONTINUE
      DO 40 L=1,NI
      R(L,II)=R(L,II)*DIRARI
40  CONTINUE
      K=K+1
      IF ((DIRARI(II)).LE.(1.+STOP)) ICTCO=ICTCO+1
      IF ((DIRARI(II)).GT.(1.+STOP)) ICTCC=0
      IF ((ICTCO.EQ.NI)) GO TO 3
      10 CONTINUE
      IF (K.LT.40) GO TO 4
      3 CONTINUE
100 CONTINUE
      DO 60 I=1,N
      DO 60 L=1,N
      A(I,L)=P(I)*A(I,L)/P(L)
60  CONTINUE
      IF (IPT.LT.1) GO TO 200
      WRITE (3,201) (P(I),I=1,N)
201 FORMAT (/10X,* THE DIAGONAL TRANSFORMATION MATRIX IS*/40X,(E20.8))
200 CONTINUE
      RETURN
      END
      ASCALE   59
      ASCALE   60
      ASCALE   61
      ASCALE   62
      ASCALE   63
      ASCALE   64
      ASCALE   65
      ASCALE   66
      ASCALE   67
      ASCALE   68
      ASCALE   69
      ASCALE   70
      ASCALE   71
      ASCALE   72
      ASCALE   73
      ASCALE   74
      ASCALE   75
      ASCALE   76
      ASCALE   77
      ASCALE   78
      ASCALE   79
      ASCALE   80
      ASCALE   81
      ASCALE   82
      ASCALE   83
      ASCALE   84
      ASCALE   85
      ASCALE   86
      ASCALE   87
      ASCALE   88
      ASCALE   89
      ASCALE   90
      ASCALE   91
      ASCALE   92
      ASCALE   93
      ASCALE   94
      ASCALE   95

```

**ORIGINAL PAGE IS
OF POOR QUALITY**

	SUBROUTINE CARD	CARD	2
	COMMON/COND/PFAD,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,	CARD	3
	1,CONTUR,NUMRS,FRPS,TRESP,MOCFL,NSCALE,SAV,CMAT,NK2,IFLAG,	CARD	4
5	1IGO,FORM,IPT,READS,MIXED,MULTRT,SCAPLT,ZOH,KOUNT	CARD	5
	INTEGER PREAD,SYSTEM,OUTPUT,FORM,CCNTUR,SAV,CMAT,READS, FRPS,TRESP	CARD	6
	INTEGER DIGITL,SCAPLT,ZOH	CARD	7
	COMMON/ACOND/DELT,FINALT,IFREQ,FFREQ,DELFRQ,GAIN1,GAIN2,M	CARD	8
	CEMPCN/LABEL/IPT,OUTPT,TITLE	CARD	9
	COMMON/SUMWRT/ISUBNAM	CARD	10
10	REAL INPT(10), OUTPT(20),TITLE(8)	CARD	11
	REAL IFREQ,M	CARD	12
	NAMFLIST/CODE/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,	CARD	13
	1,CONTUR,NUMRS,FRPS,TRESP,MOCFL,NSCALE,SAV,CMAT,NK2,IFLAG,IGO,	CARD	14
	2,FORM,IPT,READS,KOUNT,DELT,FINALT,IFREQ,FFREQ,DELFRQ,GAIN1,GAIN2,	CARD	15
15	3,MIXED,MULTRT,SCAPLT,	CARD	16
	4,ZOH,M,TSUBNAM	CARD	17
	IF (TSUBNAM .EQ. ?) WRITE (3,990)	CARD	18
	DATA PST/10HRCOCO /,PLOT/10HPLOT /	CARD	19
	990 FORMAT(1X,*CARD*)	CARD	20
20	1 FORMAT (8A10)	CARD	21
	2 FORMAT (4D12)	CARD	22
	IF (KOUNT.NE.1) GO TO 10	CARD	23
	WRITE (3,11)	CARD	24
25	11 FORMAT (/,," CONTROL PROGRAM PERMANENT FILE JWELIB,CYCLE 13, JA	CARD	25
	1INUARY 1,1976 ",/)	CARD	26
	12 CONTINLF	CARD	27
	IF (CONTUR.EQ.1,AND,KOUNT.GT.1) GO TO 5	CARD	28
	READ (1,1) TITLE	CARD	29
	IF (TITLE(11).EQ.PLOT) GO TO 300	CARD	30
30	IF (EOF(1).NE.0) STOP	CARD	31
	WRITR (3,3) TITLE	CARD	32
	3 FORMAT (1M1,8A10)	CARD	33
	IF (IFLAG .EQ.1,AND,KOUNT.GT.1) RETURN	CARD	34
	READ (1,CODE)	CARD	35
35	140 FORMAT (10X,*CONTINUOUS SYSTEM*)	CARD	36
	141 FORMAT (10X,*SAMPLED-DATA SYSTEM*)	CARD	37
	142 FORMAT (10X,*DISCRETE SYSTEM*)	CARD	38
	143 FORMAT (10X,*MIXED OPTION*)	CARD	39
40	144 FORMAT (10X,*OPEN LOOP*)	CARD	40
	145 FORMAT (10X,*CLOSED LOOP*)	CARD	41
	146 FORMAT (10X,*RCOT LOCUS*)	CARD	42
	147 FORMAT (10X,*ROOT CONTOUR*)	CARD	43
	148 FORMAT (10X,*LOAD ROUTINE INPUT*)	CARD	44
45	149 FORMAT (10X,*MATRIX ROUTINE INFUT*)	CARD	45
	150 FORMAT (10X,*CHANGE ROUTINE INFUT*)	CARD	46
	151 FORMAT (10X,*CLASS ROUTINE INFUT*)	CARD	47
	152 FORMAT (10X,*TRANSFER FUNCTIONS*)	CARD	48
	153 FORMAT (10X,*EIGENVALUES*)	CARD	49
50	154 FORMAT (10X,*FREQUENCY RESPONSES*)	CARD	50
	155 FORMAT (10X,*POWER SPECTRA*)	CARD	51
	156 FORMAT (10X,*TRANSIENT RESPONSES*)	CARD	52
	IS=DIGITL+1	CARD	53
	GO TO (170,171,172),IS	CARD	54
55	170 WRITE (3,140)	CARD	55
	GO TO 200	CARD	56
	171 WRITE (3,141)	CARD	57
	GO TO 200	CARD	58

SUBROUTINE CARD

73/74 OPT=1

FTN 4.2+7506F

01/09/76 13.59.49.

	172	WRITE (3,142)	CARD	59
60	200	IF (MIXED .EQ. 1) WRITE (3,243)	CARD	60
	GO TO (173,174,175),SYSTEM	CARD	61	
	173	WRITE (3,143)	CARD	62
	GO TO 201	CARD	63	
	174	WRITE (3,144)	CARD	64
	GO TO 201	CARD	65	
65	175	WRITE (3,145)	CARD	66
	201	GO TO (176,177,178,179),READ	CARD	67
	176	WRITE (3,146)	CARD	68
	GO TO 202	CARD	69	
	177	WRITE (3,147)	CARD	70
	GO TO 202	CARD	71	
	178	WRITE (3,148)	CARD	72
	GO TO 202	CARD	73	
	179	WRITE (3,149)	CARD	74
75	202	IF (NUMFRS .EQ. 0,AND,SYSTEM,NE.3,AND,CONTUR,EQ.0) WRITE (3,151)	CARD	75
	IF (NUMFRS .EQ. 1) WRITE (3,152)	CARD	76	
	IF (CONTUR .EQ. 1) WRITE (3,245)	CARD	77	
	IF (FRPS .EQ. 1 .OR. FRPS .EQ. -1) WRITE (3,153)	CARD	78	
	IF (FRSP .EQ. 2) WRITE (3,154)	CARD	79	
	IF (FRSP .NE. 0) WRITE (3,155)	CARD	80	
80	WRITE (3,156)	CARD	81	
	INU,MIXED,NUMFRS,IFLAG,IFREQ,NKC,OLTPUT,FORM,IGC,DELFRQ,NUC,DIGITL,	CARD	82	
	?CONTUR,READ\$,FFREQ,ZOH,IPT,MULTRT,SAV,GAIN1,N1,KOUNT,MODEL,NSCALE,	CARD	83	
	?GAIN2,N2,M	CARD	84	
85	150	FORMAT (//10X,*NW =*,I4,11X,*READ =*,I4,8X,*TRESP =*,I4,8X,	CARD	85
	1*CMAT =*,I4,8X,*DELT =*,F7.3/10X,*NY =*,I4,11X,*SYSTEM =*,	CARD	86	
	2I4,8X,*FRPS =*,I4,8X,*NK2 =*,I4,8X,*FINALT =*,F7.3/10X,	CARD	87	
	3*NU =*,I4,11X,*MIXED =*,I4,8X,*NUMFRS =*,I4,8X,*IFLAG =*,T4,	CARD	88	
	48X,*IFREQ =*,F7.3/10X,*NKC =*,I4,11X,*OUTPUT =*,I4,8X,*FORM =*,	CARD	89	
	5I4,8X,*TGO =*,I4,9X,*DELFRQ =*,F7.3/10X,*NUC =*,I-,11X,	CARD	90	
	6*DIGITL =*,I4,8X,*CONTUR =*,I4,8X,*READ3 =*,I4,8X,*FFREQ =*,	CARD	91	
	7F7.3/10X,*ZOH =*,I4,11X,*IPT =*,I4,8X,*MULTRT =*,I4,8X,	CARD	92	
	8*SAV =*,I4,8X,*GAIN1 =*,F7.3/10X,*N1 =*,I4,11X,*KOUNT =*	CARD	93	
	9I4,8X,*MODEL =*,I4,8X,*NSCALE =*,I4,8X,*GAIN2 =*,F7.3/10X,	CARD	94	
	1*N2 =*,I4,71X,*H =*,F7.3)	CARD	95	
95	IF (MULTRT.GT.0) DELT=DELT/MULTRT	CARD	96	
	IF (NY.EQ.0) GO TO 50	CARD	97	
	IF (MIXED.NE.0,AND,NY.LT.8) GO TO 50	CARD	98	
	READ (1,1)(OUTPT(I),I=1,NY)	CARD	99	
	IF (EOF(1).NE.0) STOP	CARD	100	
100	GO TO 51	CARD	101	
	51 READ (1,1)(OUTPT(I),I=1,8)	CARD	102	
	IF (EOF(1).NE.0) STOP	CARD	103	
	51 IF (NUL.EQ.0) GO TO 52	CARD	104	
	IF (MIXED.NE.0,AND,NU.LT.8) GO TO 52	CARD	105	
	READ (1,1)(INPT(I),I=1,NU)	CARD	106	
	IF (EOF(1).NE.0) STOP	CARD	107	
	GO TO 53	CARD	108	
	52 READ (1,1)(INPT(I),I=1,8)	CARD	109	
	IF (EOF(1).NE.0) STOP	CARD	110	
110	53 CONTINUE	CARD	111	
	RETURN	CARD	112	
	300 CALL COPO	CARD	113	
	STOP	CARD	114	
	E READ (1,1) TITLE	CARD	115	

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE CARD 71/74 OPT=1

FTN 4, 2+7506F 01/09/75 13.59.49.

```
115      IF (EOF(1),NE,0) GO TO 98
        WRITE (3,8) TITLE
        A FCPMAT (/RA10/)
        RETURN
98 IF (FORM,ED,0) GO TO 99
120      ND=3
      XYX=-1.
      WRITE (7) ND,XYX,XYX,XYX
99 STOP
END
```

```
CARD    116
CARD    117
CARD    118
CARD    119
CARD    120
CARD    121
CARD    122
CARD    123
CARD    124
CARD    125
```

SUBROUTINE CHANG 73/74 OPT=1

FTN 4.2+7E060 01/09/76 13:59:53.

```

      SUBROUTINE CHANG (A,B,C,H,G,F,K1,K2,K3,K4,D,W1,W2,W3,
     1M,Y,MU,MU,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      DIMENSION M1(MX,MX),M2(MX,MX),M3(MX,MX)
      COMMON/CAND/READ,SYSTEM,M,OUTPUT,NX,NY,NU,NUC,NUC,N1,N2,DIGITL,
     1CCNTUR,NUMERS,FRPS,TRESP,MODEL,NSCALE,SAV,CMAT,NK2,IFLAG,
     1IINC,FORM,IPT,READ3,NIKE,CMULTRT,SCAPLT,ZOH,KOUNT
      INTEGER READ,SYSTEM,OUTPUT,FORM,CCNTUR,SAV,CMAT,READS, FRPS,TRESP
      INTEGER DIGITL, SCAPLT, ZOH
      COMMON/ACOND/DELT,IFREQ,FFREQ,DELFRQ,GAIN1,GAIN2,M
      REAL K1, K2, K3, K4, IFREQ,M
      DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),H(MY,MX),G(MY,MX),F(MY,MU),
     1K1(MU,MX),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU)
      COMMON/RKDAT/NUMER,DENOM,GAIN,GRAPH,BLOCK,STATE,YTOV,ZTOU,YZTOK,
     1ITHINY,ITHINU,RBLOCK,NYTOV,NZTCU,NYZTOK,NYT,NUT,NY1,NU1
      REAL NUMER
      INTEGER GRAPH,BLOCK,STATE,YTOV,ZTCU,YZTOK
      DIMENSION GRAPH(20,5),RBLOCK(20,3),NUMER(20,5),DENOM(20,5),
     1XGAIN(20),STATE(20,4),ITHINY(30),ITHINU(20),YTOV(20,2),
     X_ZTOU(20,2),NYTU(8),YZTOK(20,2)
      20      C
      C   USER WRITTEN SUBROUTINE TO CHANGE SYSTEM PARAMETERS SET UP IN
      C   PREVIOUS CASE
      C
      25      COMMON/SURNAME/ ISURNAM
      IF(ISURNAM.GE.2) WRITE(3,990)
      390 FORMAT(1X,*CHANGE*)
      RETURN
      END

```

ROUTINE CLASS 73/74 OPT=1

FTN 4.2 75060

01/09/74 13.59.55.

```

SUBROUTINE CLASS (A,P,C,H,G,F,E,M1,M2,M3,
1 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
COMMON/COND/READ,SYSTEM,OUTPUT,NX,NY,NU,NKC,NUC,N1,N2,CICITL,
1 CCENTUR,NUMERS,FR,S,TRESP,MODEL,NSCALE,SAV,CMAT,NK2,IFLAG,
2 TIGO,FORM,IPT,READ3,MIXFC,MULTRT,SCAPLT,ZOH,KCOUNT
CCMNCN/ACOND/DELT,FINALTY,IFREQ,FFREQ,DELFRQ, GAIN1,GAIN2,MMH
INTEGER READ,SYSTEM,CUTFUT,FCRM,CCENTUR,SAV,CMAT,READ3, FRPS,TRESP
INTEGER DIGITL,SCAPLT,ZOH
DIMENSION A(MX,MX),B(MX,MX),C(MX,MX),H(MY,MX),G(MY,MX),F(MY,MU),
1 U(MU,MU)
DIMENSION W1(MX,MX),W2(MX,MX),W3(MX,MX)
REAL IFREQ,NUMER,MMH
COMMON/NLKDAT/NUMER,IENCM,GATA,GRAPH,BLOCK,STATE,YTOV,ZTCU,Y7TOK,
1 ITHINAY,IHTHINU,MLOCK,NYTOV,NZTOU,NXYU,NY7TOK,NXT,NYT,NUT,NY1,NUI
INTEGER GRAPH,BLOCK,STATE,YTOV,ZTCU,YZTOK
DIMENSION GRAPH(20,5),BLOCK(20,3),NUMER(20,5),CENCM(20,5),
1 XGAIN(20),STATE(20,4),IHTHINU(30),IHTHINU(20),YTOV(20,2),
X /TCU(2),2),NYNU(4) Y7TOK(20,2)
1 FORMAT (16I5)
20 FORMAT(SUBRIT/ TSUBNM
TF(IISUBNM,6F.2) WRITE(3,990)
990 FORMAT(1X,*CLASS*)
2 FORMAT (1E10.4)
NIN=3
25 NX1=NX
NY1=NY
NU1=NU
NX2=NX
NY2=NY
NU2=NU
IF ((GO.EQ.1) GO TO 224
NYTOV=0
NZTOU=0
NY7TOK=0
224 IF (MXED.EQ.1) GO TO 210
NY1=0
NY1=1
NU1=0
210 CONTINUE
30 IF (IGC.EQ.1.AND.KOUNT.GT.1) GO TO 86
DO 222 I=1,30
IHTHINU(I)=0
222 CONTINUE
DO 223 I=1,20
IHTHINU(I)=0
223 CONTINUE
READ (1,1) MLOCK,NUT
IF (.0F(1).NE.0) STOP
IF (.0F(1).EQ.0) GO TO 230
IF (INIT.EQ.0) GO TO 736
CALL GMZ (DELT,IGO,
1 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
GO TO 45
50 DC 31 I=1,MLOCK
READ (1,1) (GRAPH(I,J),J=1,5)
IF (.0F(1).NE.0) STOP
91 CONTINUE

```

CLASS 2
CLASS 3
CLASS 4
CLASS 5
CLASS 6
CLASS 7
CLASS 8
CLASS 9
CLASS 10
CLASS 11
CLASS 12
CLASS 13
CLASS 14
CLASS 15
CLASS 16
CLASS 17
CLASS 18
CLASS 19
CLASS 20
CLASS 21
CLASS 22
CLASS 23
CLASS 24
CLASS 25
CLASS 26
CLASS 27
CLASS 28
CLASS 29
CLASS 30
CLASS 31
CLASS 32
CLASS 33
CLASS 34
CLASS 35
CLASS 36
CLASS 37
CLASS 38
CLASS 39
CLASS 40
CLASS 41
CLASS 42
CLASS 43
CLASS 44
CLASS 45
CLASS 46
CLASS 47
CLASS 48
CLASS 49
CLASS 50
CLASS 51
CLASS 52
CLASS 53
CLASS 54
CLASS 55
CLASS 56
CLASS 57
CLASS 58

SUBROUTINE CLASS 73/74 OPT=1 FTN 4.2+75056 01/19/76 13.59.47.
 DO 32 I=1,NBLOCK
 READ (1,1) (BLOCK(I,J),J=1,3)
 IF (.NOT(I).NE.0) STOP
 32 CONTINUE
 DO 33 I=1,NBLOCK
 READ (1,2) (NUMBER(I,J),J=1,5)
 IF (.NOT(I).NE.0) STOP
 33 CONTINUE
 DO 34 I=1,NBLOCK
 READ (1,2) (ENCM(I,J),J=1,5)
 IF (.NOT(I).NE.0) STOP
 34 CONTINUE
 DO 35 I=1,NBLOCK
 READ (1,2) (GAIN(I),I=1,NBLOCK)
 IF (.NOT(I).NE.0) STOP
 35 CONTINUE
 DO 36 I=1,NBLOCK
 DO 37 J=1,4
 STATE (I,J)=0.0
 36 CONTINUE
 NX=0
 DO 38 I=1,NBLOCK
 STATE (I,1)=GRAPH(I,1)
 IF (.NOT(I).EQ.1.AND.BLOCK(I,3).EQ.1) GO TO 21
 IF (.NOT(BLOCK(I,2).GT.BLOCK(I,3))) GO TO 22
 IF (.NOT(BLOCK(I,2).EQ.BLOCK(I,1))) GO TO 23
 STATE (I,3)=1
 GO TO 24
 23 STATE (I,3)=2
 24 STATE (I,2)=NX+1
 STATE (I,4)=BLOCK(I,3)-1
 NX=NX+BLOCK(I,3)-1
 GO TO 20
 20 STATE (I,3)=4
 GO TO 20
 22 STATE (I,3)=3
 20 CONTINUE
 NU=1
 DO 23 K=1,NBLOCK
 IF (.NOT(K).EQ.1.AND.NU.NU=IAES(GRAPH(K,5))
 23 CONTINUE
 NX=NK+NX1
 NY=NBLOCK+NY1
 NUT=NU+NUI
 IF (.NOT(NU).EQ.1.AND.KOUNT.GT.1) GO TO 271
 READ (1,1) (ITHINY(I),I=1,NY1)
 IF (.NOT(I).NE.0) STOP
 271 FORMAT (/10X,*ITHINY*)
 IF (.NOT(I).EQ.4.AND.SYSTEM.EQ.3) GO TO 232
 IF (.NOT(I).NE.1) GO TO 243
 232 READ (1,1) (ITHINU(I),I=1,NUT)
 IF (.NOT(I).NE.0) STOP
 230 READ (1,1) NYTOV,NZTOU,NYZTOK
 IF (.NOT(I).NE.0) STOP
 IF (.NOT(NYTOV.EQ.0)) GO TO 231
 DO 231 I=1,NYTOV
 READ (1,1) (YTOV(I,J),J=1,2)
 IF (.NOT(I).NE.0) STOP

	CLASS	59
1	CLASS	60
2	CLASS	61
3	CLASS	62
4	CLASS	63
5	CLASS	64
6	CLASS	65
7	CLASS	66
8	CLASS	67
9	CLASS	68
10	CLASS	69
11	CLASS	70
12	CLASS	71
13	CLASS	72
14	CLASS	73
15	CLASS	74
16	CLASS	75
17	CLASS	76
18	CLASS	77
19	CLASS	78
20	CLASS	79
21	CLASS	80
22	CLASS	81
23	CLASS	82
24	CLASS	83
25	CLASS	84
26	CLASS	85
27	CLASS	86
28	CLASS	87
29	CLASS	88
30	CLASS	89
31	CLASS	90
32	CLASS	91
33	CLASS	92
34	CLASS	93
35	CLASS	94
36	CLASS	95
37	CLASS	96
38	CLASS	97
39	CLASS	98
40	CLASS	99
41	CLASS	100
42	CLASS	101
43	CLASS	102
44	CLASS	103
45	CLASS	104
46	CLASS	105
47	CLASS	106
48	CLASS	107
49	CLASS	108
50	CLASS	109
51	CLASS	110
52	CLASS	111
53	CLASS	112
54	CLASS	113
55	CLASS	114
56	CLASS	115

SUBROUTINE CLASS 73/74 OPT=1 FTN 4,2+75060 01/09/76 13:59:54
 115 212 CONTINUE CLASS 116
 231 IF (NZTOK.EQ.0) GO TO 211 CLASS 117
 DO 213 I=1,NZTOK
 READ (1,1) (ZTOK(I,J),J=1,2)
 IF (EOF(1).NE.0) STOP CLASS 118
 120 213 CONTINUE CLASS 119
 211 IF (NYZTOK.EQ.0) GO TO 243 CLASS 120
 DO 219 I=1,NYZTOK
 READ (1,1) (YZTOK(I,J),J=1,2)
 IF (EOF(1).NE.0) STOP CLASS 121
 125 219 CONTINUE CLASS 122
 IF (SYST-M.FQ.1) SYSTEM=2 CLASS 123
 247 CONTINUE CLASS 124
 IF (NLLOCK.EQ.0) GO TO 241 CLASS 125
 214 FORMAT (/10X,*ITHINU*) CLASS 126
 215 FORMAT (/10X,*YTOK*) CLASS 127
 217 FORMAT (/10X,*ZTOK*) CLASS 128
 220 FORMAT (/10X,*YZTOK*) CLASS 129
 GO TO 272 CLASS 130
 271 CONTINUE CLASS 131
 NY1=NYYU(1)
 NY1=NYYU(2)
 NU1=NYYU(3)
 NX1=NYYU(4)
 NYT=NYYU(5)
 NUT=NYYU(6)
 NY=NYYU(7)
 NU=NYYU(8)
 GO TO 273 CLASS 132
 140 272 NYYU(1)=NY1 CLASS 133
 NYYU(2)=NY1 CLASS 134
 NYYU(3)=NU1 CLASS 135
 NYYU(4)=NX1 CLASS 136
 NYYU(5)=NYT CLASS 137
 NYYU(6)=NUT CLASS 138
 NYYU(7)=NX CLASS 139
 NYYU(8)=NU CLASS 140
 273 CONTINUE CLASS 141
 IF (NIT.EQ.1) GO TO 340 CLASS 142
 WRITE (3,95) CLASS 143
 145 95 FORMAT (//** BLOCK DIAGRAM INPUT PARAMETERS ARE**/)
 WRITE (3,96)
 96 FORMAT (10X,* GRAPH*)
 DO 97 I=1,NBLOCK
 WRITE (3,1) (GRAPH(I,J),J=1,5)
 149 97 CONTINUE CLASS 144
 WRITE (3,98)
 98 FORMAT (/10X,*BLOCK*)
 DO 99 I=1,NBLOCK
 WRITE (3,1) (BLOCK(I,J),J=1,3)
 155 99 CONTINUE CLASS 145
 WRITE (3,81)
 81 FORMAT (10X,*NUMER*)
 DO 82 I=1,NBLOCK
 WRITE (3,2) (NUMER(I,J),J=1,5)
 170 82 CONTINUE CLASS 146
 WRITE (3,83)
 171 CLASS 147
 172 CLASS 148

SUBROUTINE CLASS 73/74 CPT=1 FTN 4.2+7E060 01/09/76 13.59.55.

 175 AT FORMAT (/10X,*DENOM *//)
 DO 44 I=1,NRLOCK
 WRITE (3,2) (DENOM(I,J),J=1,5)
 44 CONTINUE
 WRITE (3,85)
 45 FORMAT (/10X,*GAIN*)
 WRITE (3,21) (GAIN(I),I=1,NRLOCK)
 47 WRITE (3,87)
 180 WRITE (3,1)(ITHIN(Y,I),I=1,NYT)
 IF (READ.EQ.4.AND.SYSTEM.EQ.3) GO TO 341
 IF (MIXED.NE.1) GO TO 242
 341 WRITE (3,214)
 WRITE (3,1)(ITHINU(I),I=1,NUT)
 IF (NYTOV.EQ.0) GO TO 240
 WRITE (3,215)
 DO 216 I=1,NYTOV
 WRITE (3,1)(YTCV(I,J),J=1,2)
 216 CONTINUE
 240 IF (N7TCU.EQ.0) GO TO 241
 WRITE (3,217)
 DO 218 I=1,N7TCU
 WRITE (3,1)(7TOU(I,J),J=1,2)
 218 CONTINUE
 241 IF (NY7TOK.EQ.0) GO TO 242
 WRITE (3,220)
 DO 221 I=1,NY7TOK
 WRITE (3,1)(Y7TOK(I,J),J=1,2)
 221 CONTINUE
 242 CONTINUE
 IF (NRLOCK.EQ.0) RETURN
 DO 440 I=1,NRLOCK
 IF (DENOM(I,BLOCK(I,3)).EQ.1.0) GO TO 440
 NR= BLOCK(I,3)
 XX=DENOM(I,NR)
 IF (XX.NE.0.0) GO TO 442
 WRITE (3,443) I
 443 FORMAT (/10X,* LEADING COEFFICIENT OF NO.*.IS,* BLOCK IS ZERO*)
 GO TO 440
 210 442 CONTINUE
 DO 441 J=1,NR
 DENOM(I,J)= DENOM(I,J)/XX
 441 CONTINUE
 GAIN(I)= GAIN(I)/XX
 215 440 CONTINUE
 DO 30 I=1,NRLOCK
 IF (STATE(I,3).NE.4) GO TO 40
 DO 31 J=1,NR
 IF (GRAPH(I,J+1).EQ.0) GO TO 31
 C(I+N1,I)=RS(GRAPH(I,J+1))+N1=ISIGN(1,GRAPH(I,J+1))*GAIN(I)
 1*NUMER(I,1)/DENOM(I,1)
 31 CONTINUE
 IF (GRAPH(I,5).EQ.0) GO TO 30
 F(I+N1,I)=RS(GRAPH(I,5))+NU1=ISIGN(1,GRAPH(I,5))*GAIN(I)
 1*NUMER(I,1)/DENOM(I,1)
 GO TO 30
 40 IF (STATE(I,3).EQ.3) GO TO 50
 NOST1=STATE(I,4)+1

ORIGINAL PAGE IS
 OF POOR QUALITY

SUBROUTINE CLASS 73/74 OPT=1

FTN 4.2+7E 76F 01/09/76 13.59.55.

```

      NOST=NOST-1
      IF (NOST.NE.0) GO TO 45
      WRITE (3,46)
      46 FORMAT (/10X,*INCONSISTENT DATA IN CLASS*)
      GO TO 30
      47 DO 47 J=1,NOST
      H(I+NY1,STATE(I,J)+J-1+NX1)=GAIN(I)*(NUMER(I,J)-DENOM(I,J)*NUMER
      1(I,NOST))
      47 CONTINUE
      IF (STATE(I,1).EQ.1) GO TO 41
      IF (GRAPH(I,5).EQ.3) GO TO 43
      F(I+NY1,IARS(GRAPH(I,5))+NU1)=ISIGN(1,GRAPH(I,5))*GAIN(I)*NUMER
      1(I,LOCK(I,2))
      43 CONTINUE
      DO 44 J=1,NIN
      IF (GRAPH(I,J+1).EQ.0) GO TO 44
      C(I+NY1,IARS(GRAPH(I,J+1))+NY1)=-ISIGN(1,GRAPH(I,J+1))*GAIN(I)
      1*NUMER(I,BLOCK(I,2))
      44 CONTINUE
      41 CONTINUE
      GO TO 30
      50 CONTINUE
      DO 60 J=1,NIN
      IF (GRAPH(I,J+1).EQ.0) GO TO 60
      NOST=STATE(I,IABS(GRAPH(I,J+1)),4)
      IF (NOST.GE.2) GO TO 47
      WRITE (3,48)
      48 FORMAT (/10X,*DIFFERENTIATOR INPUT NOT ALLOWED*)
      GO TO 60
      47 DO 72 L=2,NOST
      H(I+NY1,STATE(I,IABS(GRAPH(I,J+1)),2)+L-1+NX1)=H(I+NY1,STATE(I,Abs
      1(GRAPH(I,J+1)),2)+L-1+NX1)+ISIGN(1,GRAPH(I,J+1))*GAIN(I)*GAIN(IABS
      2(GRAPH(I,J+1)))*(NUMER(IABS(GRAPH(I,J+1)),L-1)-NUMER(IABS(GRAPH
      3(I,J+1)),NOST)*DENOM(IABS(GRAPH(I,J+1)),L))
      72 CONTINUE
      IF (PLCK(IABS(GRAPH(I,J+1)),2)+1.LT.BLOCK(IABS(GRAPH(I,J+1)),3))
      11 GO TO 60
      H(I+NY1,STATE(I,IABS(GRAPH(I,J+1)),2)+NX1)=H(I+NY1,STATE(I,Abs
      1(GRAPH(I,J+1)),2)+NX1)+GAIN(I)*GAIN(IABS(GRAPH(I,J+1)))*ISIGN(1,GRAPH
      2(I,J+1))*(-1)*NUMER(IABS(GRAPH(I,J+1)),NOST)*DENOM(IABS(GRAPH
      3(I,J+1)),1)
      IF (GRAPH(I,IABS(GRAPH(I,J+1)),5).EQ.0) GO TO 73
      F(I+NY1,IARS(GRAPH(IABS(GRAPH(I,J+1),5))+NU1)=ISIGN(1,GRAPH(IABS
      1(GRAPH(I,J+1),5))*GAIN(I)*GAIN(IABS(GRAPH(I,J+1)))*NUMER(IABS
      2(GRAPH(I,J+1),PLCK(IABS(GRAPH(I,J+1)),2)-1)*ISIGN(1,GRAPH(I,J+1)
      3)+F(I+NY1,IARS(GRAPH(I,IABS(GRAPH(I,J+1),5))+NU1)
      73 DO 74 L=1,NIN
      C(I+NY1,IABS(GRAPH(IABS(GRAPH(I,J+1)),L+1))+NY1)=C(I+NY1,IABS(GRAPH
      1(I,IABS(GRAPH(I,J+1)),L+1))+NY1)-ISIGN(1,GRAPH(IABS(GRAPH(I,J+1)),
      ?L+1))*ISIGN(1,GRAPH(I,J+1))*GAIN(I)*GAIN(IABS(GRAPH(I,J+1)))
      74 CONTINUE
      60 CONTINUE
      30 CONTINUE
      II1=NY1+1
      II2=NLOCK+NY1
      DO 80 II=II1,II2
      C(II,II)=C(II,II)+1.0
      80 CONTINUE
      285

```

TYPE UTING CLASS 73/74 OPT=1 FTN 4.2+75060 01/09/76 13.59.55.

 240 CONTINUE
 CALL INVR (C,W1,NYT,J,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 MAT1=MX
 MAT2=MX
 MAT3=MY
 MAT4=MX
 MAT5=MX
 MAT6=MX
 245 CALL MULT (W1,H,W2,NYT,NYT,NXT,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 MAT1=MY
 MAT2=MX
 CALL MAKE (H,W2,NYT,NXT,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 MAT1=MY
 MAT2=MY
 MAT3=MU
 CALL MULT (W1,F,W2,NYT,NYT,NUT,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 MAT1=MY
 MAT2=MU
 MAT3=MX
 MAT4=MX
 305 CALL MAKE (F,W2,NYT,NUT,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 DO 200 I=1,NBLCK
 NCST=STATE(I,4)
 IF (NOST.EQ.0) GO TO 200
 DO 201 J=1,NOST
 A(STATE(I,2)+NCST-1+NX1,STATE(I,2)+J-1+NX1)=DENOM(I,J)
 IF (J.EQ.NOST) GO TO 201
 A(STATE(I,2)-1+J+NX1,STATE(I,2)+J+NX1)=1.0
 201 CONTINUE
 310 IF (IAVS(GRAPH(I,5)).EQ.0) GO TO 110
 R(STATE(I,2)+NOST-1+NX1,IAVS(GRAPH(I,5))+NU1)=ISIGN(1,GRAPH(I,5))
 315 110 CONTINUE
 DO 120 J=1,NIN
 IF (GRAPH(I,J+1).EQ.0) GO TO 120
 DO 121 K=1,NX
 A(STATE(I,2)+NOST-1+NX1,K+NX1)=A(STATE(I,2)+NCST-1+NX1,K+NY1)
 325 1+H(IAVS(GRAPH(I,J+1))+NY1,K+NX1)*ISIGN(1,GRAPH(I,J+1))
 121 CONTINUE
 DO 122 K=1,NU
 IF (F(I,K).NE.0.0) OUTPUT=3
 R(STATE(I,2)+NOST-1+NX1,K+NU1)=R(STATE(I,2)+NOST-1+NX1,K+NU1)
 330 1+F(IAVS(GRAPH(I,J+1))+NY1,K+NU1)*ISIGN(1,GRAPH(I,J+1))
 122 CONTINUE
 335 120 CONTINUE
 200 CONTINUE
 DO 230 I=1,NX
 DO 231 J=1,NX
 C(I+NX1,J+NX1)=0.0
 340 231 CONTINUE
 C(I+NX1,I+NX1)=1.0
 280 CONTINUE
 DO 232 I=1,NUT

SUPERVISOR: PLACO 7/27/74 (OPT=1)

FTN 4.2+7500.0 01/09/75 14:49:12

	DO 101 J=1,NUT	CLASS	344
	001,J=0,0	CLASS	345
160	101 CONTINU	CLASS	346
	011,I=1,0	CLASS	347
	90 CONTINU	CLASS	348
	00 522 I=1,NYT	CLASS	349
	00 522 J=1,NUT	CLASS	350
360	IF (EIT,0,1),0,0) GO TO 522	CLASS	351
	OUTOUT=3	CLASS	352
	GO TO 523	CLASS	353
	522 CONTINU	CLASS	354
	COUTOUT=1	CLASS	355
165	522 CONTINU	CLASS	356
	NUSNUT	CLASS	357
	NXENYT	CLASS	358
	NVENYT	CLASS	359
	RETURN	CLASS	360
	END	CLASS	361

```

      SUBROUTINE CNTPLR (A,B,C,H,G,F,K1,K2,K3,K4,O,W1,W2,W3,RCOTE,ROOTT, CNTPLR   ?
C 1ROTR,ROTI,SAV1,SAV2,U,V, CNTPLR   3
C 1ROTR,ROTI,7,77,U,V, CNTPLR   4
C 2WY,MV,MU,HS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6) CNTPLR   5
C
C THIS SUBROUTINE SERVES AS THE EXECUTIVE ROUTINE FOR THE CNTPLR   6
C CONTROL PROGRAM. THE CONTROL PROGRAM IS CAPABLE OF PERFORMING CNTPLR   7
C FDP LINEAR SYSTEMS. THE FOLLOWING OPERATIONS CNTPLR   8
C
C 1. ROOT LCCII AS A FUNCTION OF TWO FEEDBACK GAINS CNTPLR   9
C
C 2. DETERMINATION OF SYSTEM EIGENVALUES FOR OPEN AND CNTPLR 10
C CLOSED-LOOP SYSTEMS CNTPLR 11
C
C 3. DETERMINATION OF SYSTEM TRANSFER FUNCTIONS FOR CNTPLR 12
C ARBITRARY INPUT-OUTPUT VARIABLES CNTPLR 13
C
C 4. CALCULATION OF TABULATED FREQUENCY RESPONSES CNTPLR 14
C
C 5. CALCULATION OF TABULATED POWER SPECTRAL DENSITY CNTPLR 15
C FUNCTIONS. CNTPLR 16
C
C 6. TABULATED TIME HISTORY RESPONSE CNTPLR 17
C
C COMPUTATIONS ARE PERFORMED USING STATE VARIABLE MATRIX CNTPLR 18
C NOTATION. CNTPLR 19
C
C CORRECTION MADE BY G. NORRIS JULY 5 73 CNTPLR 20
C 7 AND SAV1, ZZ AND SAV2 ARE SAME MATRIX CNTPLR 21
C
C 30
C COMMON/COND/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL, CNTPLR 22
C 1CONTUR,NUMERS,FRPS,TRESP,MODEL,NSCALE,SAV,CMAT,NK2,IFLAG, CNTPLR 23
C 1ICG,FCRM,IPT,READT,MIXFC,NUILT,SCAPLY,ZOH,KOUNT CNTPLR 24
C INTEGER SEAD,SYSTEM,OUTPUT,FCFM,CENTUR,SAV,CMAT,READT, FRPS,TRESP CNTPLR 25
C INTEGER DIGITL,SCAPLY,70H CNTPLR 26
C COMMONCN/ACOND/ DELT,FINALT,IFREC,FFREQ,DELFRQ,GAIN1,GAIN2,MN CNTPLR 27
C REAL INPT(10), OUTPT(20),TITLE(2) CNTPLR 28
C COMMON/LABEL/IINPT,OUPUT,TITLE CNTPLR 29
C REAL IFPSO,K1,K2,K3,K4,MN CNTPLR 30
C
C DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),H(MY,MX),E(MY,MY),F(MY,MU), CNTPLR 31
C 1K1(MU,MX),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU), CNTPLR 32
C 2W1(MX,MY),W2(MX,MX),W3(MX,MX),RCOTR(MX),ROOTT(MX),ROTR(MX), CNTPLR 33
C 3ROTR(MX),U(MX),V(MX),Z(MS),ZZ(MS) CNTPLR 34
C
C DIMENSION A(15,15), B(15,10), C(15,15), H(15,15), G(15,15), CNTPLR 35
C 1 F(15,10), K1(10,15), K2(10,15), K3(10,15), K4(10,15), D(10,10) CNTPLR 36
C
C DIMENSION W1(15,15), W2(15,15), W3(15,15), ROOTT(15), ROCTT(15), CNTPLR 37
C 1 ROTR(15), ROTI(15), U(15), V(15), Z(200), ZZ(200) CNTPLR 38
C
C DIMENSION H(10,20),MM(20),P(20),ICOND(29),JCOND(29),ACOND(29) CNTPLR 39
C EQUIVALENCE (READ,ICOND(1)),(DELT,ACOND(1)) CNTPLR 40
C COMMON/RLKOAT/NUMER,DENO,GAIN,GRAPH,BLOCK,STATE,YTOV,ZTOU,YZTOK, CNTPLR 41
C 1ITHINY,IINU,NBLOCK,NYTOV,NZTCU,NXYU,NYZTOK,NXT,NYT,NUT,NY1,NU1 CNTPLR 42
C
C DIMENSION GRAPH(20,3),BLOCK(20,3),NUMER(20,5),DENOM(20,5), CNTPLR 43
C XGAIN(20),STATE(20,4),ITHINY(50),IINU(20),YTOV(20,2), CNTPLR 44
C X ZTOU(20,2),NXYU(8),YZTOK(20,2) CNTPLR 45
C
C INTEGER GRAPH,BLOCK,STATE,YTOV,ZTCU,YZTOK CNTPLR 46
C
C REAL NUMER CNTPLR 47
C COMMON/SUBWRT/ ISUB3NAME CNTPLR 48

```

ORIGINAL PAGE IS
OF POOR QUALITY

CONFIDENTIAL CONTROLLER 73/74 OPT=1

FTN 4, 2475350

31/09/75 14.02.00.

```

      DATA RCON/1CHFCOC          /
      IF(I1<UNNAME.GE.2) WRITE(1,930)
930 FORMAT(1X,*CNTRLR*)
      KOUNT=0
      ISAV=0
500 CONTINUE
      ITT=0
      DO 702 I=1,29
      JCOND(I)=0
702 CONTINUE
      DC 703 I=1,8
      ECOND(I)=0.0
703 CONTINUE
501 CONTINUE
      REINRD 8
      KOUNT=KOUNT+1
      DO 709 I=1,29
      ICOND(I)=JCOND(I)
709 CONTINUE
      CALL CARD
      IF(READ.NE.7) ISAV=0
      DC 711 I=1,29
      JCOND(I)=ICOND(I)
711 CONTINUE
      IF((CCNTUR.EQ.0).AND.(FCPM.GT.0)).AND.(ITT.EQ.0) GO TO 91
      GO TO 92
91 NPLOT=1
      WRITF (7) NPLOT
      WRITF (7) RCON,TITLE,SYSTEM,MODEL,DIGITL,SCAPLT
92 IF ((CCNTUR.EQ.0)) ITT=1
      CALL ZOT (A,B,C,H,G,F,K1,K2,K3,K4,D,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      GO TO (3),40,50,51,READ
30 CALL LOAD (A,B,C,H,G,F,K1,K2,K3,K4,D,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      GO TO 90
40 CALL MATPIX (A,B,C,H,G,F,K1,K2,K3,K4,D,W1,W2,W3,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      GO 45 I=1,29
45 JCOND(I)=ICOND(I)
50 GO TO 60
50 IF ((ISAV.EQ.0)) GO TO 51
      CALL PDTSC (A,B,C,H,G,F,K1,K2,K3,K4,D,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
51 CALL CHANGE (A,B,C,H,G,F,K1,K2,K3,K4,D,W1,W2,W3,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF ((MIXED.EQ.0.AND.ISAV.EQ.0)) GO TO 555
      GO TO 60
55 IGO =0
555 CONTINUE
      CALL CLASS      (A,B,C,H,G,F,D,W1,W2,W3,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
60 CONTINUE
      IF (READ3.EQ.0) JCOND(1)=3
      IF (SAV.EQ.0) ISAV=1
      CNTRLR 59
      CNTRLR 60
      CNTRLR 61
      CNTRLR 62
      CNTRLR 63
      CNTRLR 64
      CNTRLR 65
      CNTRLR 66
      CNTRLR 67
      CNTRLR 68
      CNTRLR 69
      CNTRLR 70
      CNTRLR 71
      CNTRLR 72
      CNTRLR 73
      CNTRLR 74
      CNTRLR 75
      CNTRLR 76
      CNTRLR 77
      CNTRLR 78
      CNTRLR 79
      CNTRLR 80
      CNTRLR 81
      CNTRLR 82
      CNTRLR 83
      CNTRLR 84
      CNTRLR 85
      CNTRLR 86
      CNTRLR 87
      CNTRLR 88
      CNTRLR 89
      CNTRLR 90
      CNTRLR 91
      CNTRLR 92
      CNTRLR 93
      CNTRLR 94
      CNTRLR 95
      CNTRLR 96
      CNTRLR 97
      CNTRLR 98
      CNTRLR 99
      CNTRLR 100
      CNTRLR 101
      CNTRLR 102
      CNTRLR 103
      CNTRLR 104
      CNTRLR 105
      CNTRLR 106
      CNTRLR 107
      CNTRLR 108
      CNTRLR 109
      CNTRLR 110
      CNTRLR 111
      CNTRLR 112
      CNTRLR 113
      CNTRLR 114
      CNTRLR 115

```

ROUTINE ENTRY 1374 RPT=1 FTN 4.2+75 06.0 01/09/76 14.02.00.

```

115      IF (ICONTUR,EQ.1.AND.KOUNT.GT.1) GO TO 69
      CALL SPRT (A,B,C,H,G,F,K1,K2,K3,K4,D,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      GO TO 120
120      CALL WDISC (A,B,C,H,G,F,K1,K2,K3,K4,D,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      JDNF(1)=0
      7L CONTINUE
      CALL SITUP (J,M,MM,P,A,R,C,H,G,F,K1,K2,K3,K4,D,W1,W2,W3,
      1 R00TR,
      2 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF (TRSP,GT.0) NPLOT=1
      IF (TRSP,GT.0) NPLOT=TRESP
      IF (TRSP,NE.0) NPLOT=NY*NU
      IF (TRPS,GT.0.AND.TRESP.GT.0) NPLCT=NY*NU+1
      IF (TRPS,NE.0.AND.TRESP.GT.0) NPLCT=NY*NU+TRESP
      IF (ISYSTEM,EQ.3) NPLOT=1
      IF (CFRM,GT.0.AND.CONTR.EQ.0) WRITE (7) NPLOT
      IF (IMULTR,NE.0) GO TO 101
      IF (INSTEM,NE.3) GO TO 50
      CALL ROOT (A,B,C,H,G,F,K1,K2,K3,K4,D,W1,W2,W3,ROOTR,RCOTI,U,V,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      GO TO 90
      80 CONTINUE
      WRITE (7,81)
140      81 FORMAT (//10X,*THE EIGEN VALUES OF THE SYSTEM ARE*//20X,
      *REAL PART*,15X,* IMAGINARY PART*//)
      CALL EIGEN (NX,W1,W2,W3,ROOTR,RCOTI,U,V,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF (IFLAG,FE.0.AND.CONTR,FE.1) GO TO 499
      IF (ICONTR,FE.1) GO TO 501
      GO TO 812
      499 IF (FOPM,EQ.0) GO TO 500
      NC=1
      XYX=-1.0
      WRITE (7) ND,XYX,XYX
      GO TO 800
      802 CONTINUE
      CALL CPMT (7,RCOTR,ROOTI,NX,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      WRITE (7,82)
155      82 FORMAT (//10X,*THE COEFFICIENTS OF THE CHARACTERISTIC EQUATION ARE*
      *URED FROM THE LOWEST POWER OF S*//)
      NX=NX+1
      WRITE (7,A3) (7(I),I=1,NX)
      A3 FORMAT (E20.0)
      83 FORMAT (E20.0)
      90 CONTINUE
      IF (KUMPS,EQ.1) GO TO 100
      NN=NX
      C      CALL NHRATP (NN,A,B,C,H,G,F,D,ROOTR,ROOTI,ROTP,ROTI,Z,V,
      CALL NHRATR (NN,A,B,C,H,G,F,D,RCOTR,RCOTI,ROTP,RCOTI,U,V,
      C      1W1,W2,W3,SAV1,SAV2,
      1W1,W2,W3,7,77,
      2MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      100 CONTINUE
      IF (TRESP,EQ.0) GO TO 507
      101 CONTINUE

```

SUBROUTINE CNTRLR 73/74 OPT=1 FTN 4.2+75060 01/09/76 14.02.00.

```
CALL      THIST (A,B,C,H,F,W1,W2,W3,ROOTR,ROCTI,U,K1,D,Z,V, CNTPLR 173
CALL      THEST (A,B,C,H,F,W1,W2,W3,ROOTR,ROCTI,U,K1,C,ROTP, CNTPLR 174
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6) CNTPLR 175
175    507 IF ((CAPLT,F0,2) JCOND(28)=1 CNTPLR 176
      IF ((FLAG>EQ.0) GO TO 100 CNTPLR 177
      GO TO 501 CNTPLR 178
      END CNTPLR 179
```

SUBROUTINE CPMT 73/74 CPT=1 FTN 4.2676060 01/09/76 14.02.41.
 * SUBROUTINE CPMT(C,ROOTR,ROOTI,N,
 * 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 * COMPLEX A,B,C,D,E,F
 * DIMENSION A(25),B(25),C(MX),D(MY),ROOTR(MX),D(25),E(25)
 * COMMON/URWPI/ ISURNAM
 * IF(ISURNAM.GE.2) WRITE(13,990)
 * 990 FORMAT(1X,*CPMT*)
 * IF (IN.GT.1) GO TO 10
 * C(1)=+ROOTR(1)
 * C(2)=1.J
 * RETURN
 * 10 CONTINUE
 * A(1)=CMPLX(-ROOTR(1),ROOTI(1))
 * A(2)=CMPLX(1.0,0.0)
 * NX=2
 * DO 4 II=2,N
 * NY=NX+1
 * DO 1 I=1,NY
 * D(I)=CMPLX(0.0,0.0)
 * 1 CONTINUE
 * 2 D(1)=CMPLX(-ROOTR(1),ROOTI(1))
 * 2 D(2)=CMPLX(1.0,0.0)
 * DO 3 I=1,2
 * DO 3 J=1,NX
 * K=I+J-1
 * D(K)=A(I)*B(J)+D(K)
 * 3 CONTINUE
 * NX=NX+1
 * DO 5 JJ=1,NX
 * A(JJ)=D(JJ)
 * 5 CONTINUE
 * 6 CONTINUE
 * DO 7 I=1,NX
 * E(I)=D(I)
 * 7 CONTINUE
 * DO 8 I=1,NX
 * C(I)=REAL(D(I))
 * 8 CONTINUE
 * RETURN
 * END

ORIGINAL PAGE IS
 OF POOR QUALITY

```

      SUBROUTINE EAT (T,A,M1,M2,M3,C,N,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
C
C THIS SUBROUTINE COMPUTES THE TRANSITION MATRIX AND ITS INTEGRAL.
C THE SERIES IS TRUNCATED WHEN THE LARGEST ELEMENT OF THE LAST TERM
C IS LESS THAN 1.E-07 TIMES THE SMALLEST ELEMENT OF THE S
C SERIES. WRITTEN BY R. MAINE 8/17/71
C
      DIMENSION A(MX,MX),W1(MX,MX),W2(MX,MX),W3(MX,MX),C(MX,MX)
      COMMON/COND/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,
1CCNTUR,NUMERS,FRPS,TRESP,MOCFL,NSCALE,SAV,CHAT,NK2,IFLAG,
1IGO,FORM,IPT,READ3,MIXED,MULTRT,SCAPLT,ZOM,KCUNT
      INTEGER READ,SYSTEM,OUTPUT,FORM,CCNTUR,SAV,CHAT,READ3, FRPS,TRESP
      INTEGER DIGITL
      COMMON/SUBRWT/ ISUBRNM
      IF (ISUBRNM.GE.2) WRITE(1,990)
990 FORMAT(1X,*EAT*)
      MAT1=MX
      MAT2=MX
      MAT3=MX
      MAT4=MX
      MAT5=MX
      MAT6=MX
      NT=24
      CALL ZCT1(M1,N,N,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL MAKE (W2,M1,N,N,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      DO 1 I=1,N
      W1(I,I)=1.0
1  CONTINUE
      CALL MAKE (W3,M1,N,N,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      G=1.0
      W3MAX=.1.E+50
      T= T/R
      DO 7 I=1,NT
      D0=I
      G=G*T/PR
      W1MIN=.1.E+50
      W2MIN=.1.E+50
      DO 30 J=1,NX
      DO 30 K=1,NX
      IF (W1(J,K).NE.0.0.AND.ABS(W1(J,K)).LT. W1MIN) W1MIN=ABS(W1(J,K))
      IF (W2(J,K).NE.0.0.AND.ABS(W2(J,K)).LT. W2MIN) W2MIN=ABS(W2(J,K))
      30  CONTINUE
      W3MAX= W3MAX*T/BB
      CALL ADD (1.0,W2,G,W3,W2,N,N,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL MULT (A,W3,C,N,N,K,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL MAKE (W3,C,N,N,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      W3MAX =0.0
      DO 40 J=1,NX
      DO 40 K=1,NX
      IF (ABS(W3(J,K)).GT. W3MAX) W3MAX= (ABS(W3(J,K)))
      EAT   2
      EAT   3
      EAT   4
      EAT   5
      EAT   6
      EAT   7
      EAT   8
      EAT   9
      EAT  10
      EAT  11
      EAT  12
      EAT  13
      EAT  14
      EAT  15
      EAT  16
      EAT  17
      EAT  18
      EAT  19
      EAT  20
      EAT  21
      EAT  22
      EAT  23
      EAT  24
      EAT  25
      EAT  26
      EAT  27
      EAT  28
      EAT  29
      EAT  30
      EAT  31
      EAT  32
      EAT  33
      EAT  34
      EAT  35
      EAT  36
      EAT  37
      EAT  38
      EAT  39
      EAT  40
      EAT  41
      EAT  42
      EAT  43
      EAT  44
      EAT  45
      EAT  46
      EAT  47
      EAT  48
      EAT  49
      EAT  50
      EAT  51
      EAT  52
      EAT  53
      EAT  54
      EAT  55
      EAT  56
      EAT  57
      EAT  58

```

ROUTINE	LINE	OPT	FTN	DATE	TIME
MAIN	1		FTN 4.2+76.06	01/09/71	14.02.44.
FC	40	CONTINUE	EAT	59	
		W3MAX= W3MAX*G	EAT	60	
		CALL ADD (3,0, W1, G, W3, W1, N, N,	EAT	61	
		1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	EAT	62	
		IF (W3MAX.LT.W2MIN* 1.0E-03 .AND.W3MAX.LT.W1MIN* 1.0E-03) GO TO 70	EAT	63	
	71	CONTINUE	EAT	64	
		W3IT (3,1000) W1MIN, W3MAX, W2MIN, W2MAX1	EAT	65	
FE	1000	FCMATEC ERROR IN ZAT*.5X,W1MIN =*,F15.6,5X,W3MAX =*,F15.6,/* *,	EAT	66	
		117X,W2MIN =*,E15.6,GX,W3MAX1 =*,E15.6)	EAT	67	
	72	CONTINUE	EAT	68	
		DO 90 K=1,3	EAT	69	
		CALL MAKF (W3, W1, N, N,	EAT	70	
		1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	EAT	71	
		CALL MULT (W1, W3, G, N, N, N,	EAT	72	
		1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	EAT	73	
		CALL MAKF (W1, G, N, N,	EAT	74	
		1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	EAT	75	
		DO 12 J=1,N	EAT	76	
		W3(J,J)= W3(J,J)+1.	EAT	77	
	80	CONTINUE	EAT	78	
		CALL MULT (W2, W3, G, N, N, N,	EAT	79	
		1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	EAT	80	
		CALL MAKF (W2, G, N, N,	EAT	81	
		1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	EAT	82	
	90	CONTINUE	EAT	83	
		TE= T**.	EAT	84	
		W3IT (3,51) I	EAT	85	
FS	51	FORMAT (/* THE TRANSITION MATRIX *,15,* TERMS*)/	EAT	86	
		CALL SPITI (W1,NX,NX,	EAT	87	
		1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	EAT	88	
		RETURN	EAT	89	
		END	EAT	90	

```

      SUBROUTINE EIGEN (W1,W2,W3,ROCTR,ROOTI,ROTR,ROTI,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6),
2CMMCN/COND/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGIT,
3CONTUR,NUMHRS,FRPS,TRESP,MODEL,NSCALE,S,SAV,CMAT,NK2,IFLAG,
4LIGO,FPM,IPT,PEAT3,MIXID,MULTRT,SCAPLT,ZOH,KOUNT
5INTEGER READ,SYSTEM,OUTPUT,FORM,CONTUR,SAV,CMAT,READ*, FRPS,TRESP>
6INTEGER DIGIT,SCAPLT,ZOH
7DIMENSION M(10,20),M(20),P(20),KD(25)
8DIMENSION W1(MX,MX),W2(MX,MX),W3(MX,MX)
9DIMENSION ROOTI(MX),ROOTI(MX1),ROTR(MX),ROTI(MX)
10DIMENSION NSCALE/MUIM/MX,MY,MU,MS
11C
12C THIS SUBROUTIN FINDS THE EIGENVALUES OF THE INPUT MATRIX (W1)
13C
14C NSCALE=0 NO PRECONDITIONING OF THE INPUT MATRIX IS DONE
15C
16C NSCALE=1 SCALF IS CALLED TO SCALE THE INPUT MATRIX BY A
17C DIAGONAL SIMILARITY TRANSFORMATION. THEN THE
18C EIGENVALUES OF THE IRREDUCIBLE SUBMATRICES ARE
19C DETERMINED.
20C NSCALE=2 REDUCE IS CALLED TO DETERMINE THE IRREDUCIBLE
21C SUBMATRICES OF W1, THEN THE EIGENVALUES OF THE
22C INDIVIDUAL SUBMATRICES ARE DETERMINED.
23C
24C
25C
26C COMMON/SUBWRIT/ ISUBNAM
27C IF(ISUBNAM.GE.2) WRITE(3,990)
28C 990 FORMAT(1X,*EIGEN*)
29C IFRNT=0
30C IF (NSCALE.EQ.0) GO TO 50
31C IF (NSCALE.EQ.2) GO TO 100
32C CALL ASCALE (N,J,M,M,P,W1,W2,W3,ROCTR,
33C 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
34C GO TO 101
35C 100 CALL REDUCE (N,J,M,M,W1,W2,W3,
36C 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
37C GO TO 101
38C 50 DO 51 I=1,N
39C   W1(I,I)=I
40C 51 CONTINUE
41C   MM=I
42C   J=1
43C 101 CCNTINUE
44C   KRUNT=0
45C   DO 200 I=1,J
46C     N=MM*I
47C     DO 110 K=1,NS
48C       DO 110 L=1,NS
49C         W2(K,L)=W1(M(I,K),M(I,L))
50C
51C 110 CONTINUE
52C   CALL HESSEN (W2,NS,ROTR,
53C   1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
54C   CALL OREIG (NS,ROTR,ROTI,IPONT,W2,
55C   1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
56C   IF(SYSTEM.EQ.3) GO TO 3
57C   IF(CCNTUP.EQ.11) GO TO 3
58C   GO TO 5

```

SUBROUTINE EIGEN

7374 CPT#1

FTN 4.267E0HC

01/09/76 14.02.01.

```

      IF(FCRM,F0,3) GO TO 5
      J1=0
      DO 413 J1=1,N5
      IF(PCTI(J1),LT,0.) GO TO 413
      J1=J1+1
      KC(J1)=J1
      +1P CONTINUE
      WRITE(*,13,(ROTR(KD(K)),PCTI(KC(K)),K=1,J3))
      IF(FCRM,F0,2) GO TO 7
      WRITE(3,300) (RCTR(II),ROTR(II),II=1,N5)
      7 CONTINUE
      800 FORMAT (/12F10.8)
      DO 120 K=1,N5
      R00TR(KRUNIT+K)=RCTR(K)
      P00T(KRUNIT+K)=PCTI(K)
      120 CONTINUE
      KRUNIT=KRUNIT+N5
      200 CONTINUE
      RETURN
      END

```

```

      EIGEN    9
      EIGEN   10
      EIGEN   11
      EIGEN   12
      EIGEN   13
      EIGEN   14
      EIGEN   15
      EIGEN   16
      EIGEN   17
      EIGEN   18
      EIGEN   19
      EIGEN   20
      EIGEN   21
      EIGEN   22
      EIGEN   23
      EIGEN   24
      EIGEN   25
      EIGEN   26
      EIGEN   27
      EIGEN   28
      EIGEN   29
      EIGEN   30

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE FRORSP 7374 OPT=1

FTN 4.2+75060 01/09/76 14.12.5F.

```

SUBROUTINE FRORSP (INUM,NN,CATA,IPOD,RCOTR,ROTR,ROTI,SAV1,
1 SAV2,INY,INU,
2 MX,MY,MI,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
3 REAL IFFEQ,M
4 DIMENSION ROOTR(MX),ROTR(MX),ROTI(MX),SAV1(MS),SAV2(MS)
5 COMPLEX RN1,RN2,FCL,RDP
6 INTEGER TWO,FOUR
7 COMMON/LARL/INPT,OUTPT,TITLE
8 REAL INPT(10), OUTPT(20),TITLE(8)
9 COMMON/ONDZ/READ,SYSTEM,OUTPUT,NX,NY,NU,NKC,NUC,N1,N2,DIGITL,
10 1CCNTL,NUMERS,FRPS,TRESP,MOPFL,VSCALE,SAV,CHAT,NK2,IFLAG,
11 1IGO,FORM,INPT,READ3,MIXED,MULTRT,SCAPLT,ZOH,KOUNT
12 INTEGER READ,SYSTEM,OUTPUT,FORM,CONTUR,SAV,CHAT,READ3, FRPS,TRESP
13 INTEGER DIGITL,SCAPLT, ZOH
14 COMMON/COND/ DELT,FINALT,IFREQ,FFRE0,DELFRE,GAIN1,GAIN2,M
15 COMMON/SUBP/ TSU3NA
16 REAL PST
17 DATA TWO/2/,FOUR/4/
18 DATA PST/10MREQ /
19 20 IF(TSU3NA.EQ.2) WRITE(3,930)
21 FORMAT(1X,*FRORSP*)
22 IF((FORM.GT.0).OR.WRITE(7)PST,TITLE,SYSTEM,MODEL,DIGITL,SCAPLT
23 J=1
24 IF ((IMOD,FO,2).GT.0) GO TO 70
25 IF ((CIC(TTL,NE,0).GT.0) GO TO 200
26 WRITE(1,71) OUTPT(INY),INPT(INU)
27 71 FORMAT(//5X,A10//*A10,*FREQUENCY RESPONSE S-PLANE//*
28 *A10,*AMPLITUDE RATIO*,9X,*PHASE ANGLE/* RAD/SEC*,12X,*DB*,17X,*2DEGREES*/
29 72 GO TO 76
30 76 WRITE(1,76) OUTPT(INY),INFT(INU)
31 77 FORMAT(//5X,A10//*A10,* FREQUENCY RESPONSE W-PLANE//*
32 *A10,*S-PLANE//* FREQUENCY FREQUENCY AMPLITUDE RATIO
33 *PHASE ANGLE*/12X,* RAD/SEC*,12X,*DB*,17X,*DEGREES*/
34 78 IF ((FORM.GT.0)) WRITE(7)TWO,OUTPT(INY),INPT(INU)
35 79 GO TO 72
36 80 IF ((CIC(TL,NE,0).GT.0) GO TO 201
37 WRITE(3,73) OUTPT(INY),INFT(INU),OUTPT(INY),OUTPT(INY-1)
38 73 FORMAT(3,73) CUTPT(INY),INFT(INU),OUTPT(INY),CUTPT(INY-1)
39 81 FORMAT(//6X,*S-PLANE*,A10,*//,A10,* FREQUENCY RESPONSE*3X,A10,*/*
40 *A10,* FREQUENCY RESPONSE//* W-PLANE S-PLANE//* FREQUENCY
41 82 AMPLITUDE RATIO*9X,*PHASE ANGLE AMPLITUDE RATIO*9X,*PHASE ANGLE//*
42 * RAD/SEC*10X,*DB*17X,*DEGREES*,13X,*DB*,18X,*DEGREES*/
43 83 77 IF ((FORM.GT.0)) WRITE(7)FOUR,OUTPT(INY),INPT(INU),OUTPT(INY),OUTPT
44 84 1(INY-1)
45 72 CONTINUE
46 85 FREQ=1/1.15
47 IF ((DIGITL.EQ.0).OR.INY.GT.1.ORINU.GT.1.OR.FRPS.EQ.-1) GO TO 91
48 IF ((IFFEQ.EQ.0) GO TO 90
49 IFFEQ= TAN(IFREQ*DELT*.5)
50 FFREQ= TAN(FFREQ*DELT*.5)
51 90 GO TO 91
52 91 53
54 92 55
55 93 56
56 94 57
57 95 58
58 96 59
59 97 60
60 98 61
61 99 62
62 100 63
63 101 64
64 102 65
65 103 66
66 104 67
67 105 68
68 106 69
69 107 70
70 108 71
71 109 72
72 110 73
73 111 74
74 112 75
75 113 76
76 114 77
77 115 78
78 116 79
79 117 80
80 118 81
81 119 82
82 120 83
83 121 84
84 122 85
85 123 86
86 124 87
87 125 88
88 126 89
89 127 90
90 128 91
91 129 92
92 130 93
93 131 94
94 132 95
95 133 96
96 134 97
97 135 98
98 136 99
99 137 100
100 138 101
101 139 102
102 140 103
103 141 104
104 142 105
105 143 106
106 144 107
107 145 108
108 146 109
109 147 110
110 148 111
111 149 112
112 150 113
113 151 114
114 152 115
115 153 116
116 154 117
117 155 118
118 156 119
119 157 120
120 158 121
121 159 122
122 160 123
123 161 124
124 162 125
125 163 126
126 164 127
127 165 128
128 166 129
129 167 130
130 168 131
131 169 132
132 170 133
133 171 134
134 172 135
135 173 136
136 174 137
137 175 138
138 176 139
139 177 140
140 178 141
141 179 142
142 180 143
143 181 144
144 182 145
145 183 146
146 184 147
147 185 148
148 186 149
149 187 150
150 188 151
151 189 152
152 190 153
153 191 154
154 192 155
155 193 156
156 194 157
157 195 158
158 196 159
159 197 160
160 198 161
161 199 162
162 200 163
163 201 164
164 202 165
165 203 166
166 204 167
167 205 168
168 206 169
169 207 170
170 208 171
171 209 172
172 210 173
173 211 174
174 212 175
175 213 176
176 214 177
177 215 178
178 216 179
179 217 180
180 218 181
181 219 182
182 220 183
183 221 184
184 222 185
185 223 186
186 224 187
187 225 188
188 226 189
189 227 190
190 228 191
191 229 192
192 230 193
193 231 194
194 232 195
195 233 196
196 234 197
197 235 198
198 236 199
199 237 200
200 238 201
201 239 202
202 240 203
203 241 204
204 242 205
205 243 206
206 244 207
207 245 208
208 246 209
209 247 210
210 248 211
211 249 212
212 250 213
213 251 214
214 252 215
215 253 216
216 254 217
217 255 218
218 256 219
219 257 220
220 258 221
221 259 222
222 260 223
223 261 224
224 262 225
225 263 226
226 264 227
227 265 228
228 266 229
229 267 230
230 268 231
231 269 232
232 270 233
233 271 234
234 272 235
235 273 236
236 274 237
237 275 238
238 276 239
239 277 240
240 278 241
241 279 242
242 280 243
243 281 244
244 282 245
245 283 246
246 284 247
247 285 248
248 286 249
249 287 250
250 288 251
251 289 252
252 290 253
253 291 254
254 292 255
255 293 256
256 294 257
257 295 258
258 296 259
259 297 260
260 298 261
261 299 262
262 300 263
263 301 264
264 302 265
265 303 266
266 304 267
267 305 268
268 306 269
269 307 270
270 308 271
271 309 272
272 310 273
273 311 274
274 312 275
275 313 276
276 314 277
277 315 278
278 316 279
279 317 280
280 318 281
281 319 282
282 320 283
283 321 284
284 322 285
285 323 286
286 324 287
287 325 288
288 326 289
289 327 290
290 328 291
291 329 292
292 330 293
293 331 294
294 332 295
295 333 296
296 334 297
297 335 298
298 336 299
299 337 300
300 338 301
301 339 302
302 340 303
303 341 304
304 342 305
305 343 306
306 344 307
307 345 308
308 346 309
309 347 310
310 348 311
311 349 312
312 350 313
313 351 314
314 352 315
315 353 316
316 354 317
317 355 318
318 356 319
319 357 320
320 358 321
321 359 322
322 360 323
323 361 324
324 362 325
325 363 326
326 364 327
327 365 328
328 366 329
329 367 330
330 368 331
331 369 332
332 370 333
333 371 334
334 372 335
335 373 336
336 374 337
337 375 338
338 376 339
339 377 340
340 378 341
341 379 342
342 380 343
343 381 344
344 382 345
345 383 346
346 384 347
347 385 348
348 386 349
349 387 350
350 388 351
351 389 352
352 390 353
353 391 354
354 392 355
355 393 356
356 394 357
357 395 358
358 396 359
359 397 360
360 398 361
361 399 362
362 400 363
363 401 364
364 402 365
365 403 366
366 404 367
367 405 368
368 406 369
369 407 370
370 408 371
371 409 372
372 410 373
373 411 374
374 412 375
375 413 376
376 414 377
377 415 378
378 416 379
379 417 380
380 418 381
381 419 382
382 420 383
383 421 384
384 422 385
385 423 386
386 424 387
387 425 388
388 426 389
389 427 390
390 428 391
391 429 392
392 430 393
393 431 394
394 432 395
395 433 396
396 434 397
397 435 398
398 436 399
399 437 400
400 438 401
401 439 402
402 440 403
403 441 404
404 442 405
405 443 406
406 444 407
407 445 408
408 446 409
409 447 410
410 448 411
411 449 412
412 450 413
413 451 414
414 452 415
415 453 416
416 454 417
417 455 418
418 456 419
419 457 420
420 458 421
421 459 422
422 460 423
423 461 424
424 462 425
425 463 426
426 464 427
427 465 428
428 466 429
429 467 430
430 468 431
431 469 432
432 470 433
433 471 434
434 472 435
435 473 436
436 474 437
437 475 438
438 476 439
439 477 440
440 478 441
441 479 442
442 480 443
443 481 444
444 482 445
445 483 446
446 484 447
447 485 448
448 486 449
449 487 450
450 488 451
451 489 452
452 490 453
453 491 454
454 492 455
455 493 456
456 494 457
457 495 458
458 496 459
459 497 460
460 498 461
461 499 462
462 500 463
463 501 464
464 502 465
465 503 466
466 504 467
467 505 468
468 506 469
469 507 470
470 508 471
471 509 472
472 510 473
473 511 474
474 512 475
475 513 476
476 514 477
477 515 478
478 516 479
479 517 480
480 518 481
481 519 482
482 520 483
483 521 484
484 522 485
485 523 486
486 524 487
487 525 488
488 526 489
489 527 490
490 528 491
491 529 492
492 530 493
493 531 494
494 532 495
495 533 496
496 534 497
497 535 498
498 536 499
499 537 500
500 538 501
501 539 502
502 540 503
503 541 504
504 542 505
505 543 506
506 544 507
507 545 508
508 546 509
509 547 510
510 548 511
511 549 512
512 550 513
513 551 514
514 552 515
515 553 516
516 554 517
517 555 518
518 556 519
519 557 520
520 558 521
521 559 522
522 560 523
523 561 524
524 562 525
525 563 526
526 564 527
527 565 528
528 566 529
529 567 530
530 568 531
531 569 532
532 570 533
533 571 534
534 572 535
535 573 536
536 574 537
537 575 538
538 576 539
539 577 540
540 578 541
541 579 542
542 580 543
543 581 544
544 582 545
545 583 546
546 584 547
547 585 548
548 586 549
549 587 550
550 588 551
551 589 552
552 590 553
553 591 554
554 592 555
555 593 556
556 594 557
557 595 558
558 596 559
559 597 560
560 598 561
561 599 562
562 600 563
563 601 564
564 602 565
565 603 566
566 604 567
567 605 568
568 606 569
569 607 570
570 608 571
571 609 572
572 610 573
573 611 574
574 612 575
575 613 576
576 614 577
577 615 578
578 616 579
579 617 580
580 618 581
581 619 582
582 620 583
583 621 584
584 622 585
585 623 586
586 624 587
587 625 588
588 626 589
589 627 590
590 628 591
591 629 592
592 630 593
593 631 594
594 632 595
595 633 596
596 634 597
597 635 598
598 636 599
599 637 600
600 638 601
601 639 602
602 640 603
603 641 604
604 642 605
605 643 606
606 644 607
607 645 608
608 646 609
609 647 610
610 648 611
611 649 612
612 650 613
613 651 614
614 652 615
615 653 616
616 654 617
617 655 618
618 656 619
619 657 620
620 658 621
621 659 622
622 660 623
623 661 624
624 662 625
625 663 626
626 664 627
627 665 628
628 666 629
629 667 630
630 668 631
631 669 632
632 670 633
633 671 634
634 672 635
635 673 636
636 674 637
637 675 638
638 676 639
639 677 640
640 678 641
641 679 642
642 680 643
643 681 644
644 682 645
645 683 646
646 684 647
647 685 648
648 686 649
649 687 650
650 688 651
651 689 652
652 690 653
653 691 654
654 692 655
655 693 656
656 694 657
657 695 658
658 696 659
659 697 660
660 698 661
661 699 662
662 700 663
663 701 664
664 702 665
665 703 666
666 704 667
667 705 668
668 706 669
669 707 670
670 708 671
671 709 672
672 710 673
673 711 674
674 712 675
675 713 676
676 714 677
677 715 678
678 716 679
679 717 680
680 718 681
681 719 682
682 720 683
683 721 684
684 722 685
685 723 686
686 724 687
687 725 688
688 726 689
689 727 690
690 728 691
691 729 692
692 730 693
693 731 694
694 732 695
695 733 696
696 734 697
697 735 698
698 736 699
699 737 700
700 738 701
701 739 702
702 740 703
703 741 704
704 742 705
705 743 706
706 744 707
707 745 708
708 746 709
709 747 710
710 748 711
711 749 712
712 750 713
713 751 714
714 752 715
715 753 716
716 754 717
717 755 718
718 756 719
719 757 720
720 758 721
721 759 722
722 760 723
723 761 724
724 762 725
725 763 726
726 764 727
727 765 728
728 766 729
729 767 730
730 768 731
731 769 732
732 770 733
733 771 734
734 772 735
735 773 736
736 774 737
737 775 738
738 776 739
739 777 740
740 778 741
741 779 742
742 780 743
743 781 744
744 782 745
745 783 746
746 784 747
747 785 748
748 786 749
749 787 750
750 788 751
751 789 752
752 790 753
753 791 754
754 792 755
755 793 756
756 794 757
757 795 758
758 796 759
759 797 760
760 798 761
761 799 762
762 800 763
763 801 764
764 802 765
765 803 766
766 804 767
767 805 768
768 806 769
769 807 770
770 808 771
771 809 772
772 810 773
773 811 774
774 812 775
775 813 776
776 814 777
777 815 778
778 816 779
779 817 780
780 818 781
781 819 782
782 820 783
783 821 784
784 822 785
785 823 786
786 824 787
787 825 788
788 826 789
789 827 790
790 828 791
791 829 792
792 830 793
793 831 794
794 832 795
795 833 796
796 834 797
797 835 798
798 836 799
799 837 800
800 838 801
801 839 802
802 840 803
803 841 804
804 842 805
805 843 806
806 844 807
807 845 808
808 846 809
809 847 810
810 848 811
811 849 812
812 850 813
813 851 814
814 852 815
815 853 816
816 854 817
817 855 818
818 856 819
819 857 820
820 858 821
821 859 822
822 860 823
823 861 824
824 862 825
825 863 826
826 864 827
827 865 828
828 866 829
829 867 830
830 868 831
831 869 832
832 870 833
833 871 834
834 872 835
835 873 836
836 874 837
837 875 838
838 876 839
839 877 840
840 878 841
841 879 842
842 880 843
843 881 844
844 882 845
845 883 846
846 884 847
847 885 848
848 886 849
849 887 850
850 888 851
851 889 852
852 890 853
853 891 854
854 892 855
855 893 856
856 894 857
857 895 858
858 896 859
859 897 860
860 898 861
861 899 862
862 900 863
863 901 864
864 902 865
865 903 866
866 904 867
867 905 868
868 906 869
869 907 870
870 908 871
871 909 872
872 910 873
873 911 874
874 912 875
875 913 876
876 914 877
877 915 878
878 916 879
879 917 880
880 918 881
881 919 882
882 920 883
883 921 884
884 922 885
885 923 886
886 924 887
887 925 888
888 926 889
889 927 890
890 928 891
891 929 892
892 930 893
893 931 894
894 932 895
895 933 896
896 934 897
897 935 898
898 936 899
899 937 900
900 938 901
901 939 902
902 940 903
903 941 904
904 942 905
905 943 906
906 944 907
907 9
```

SUBROUTINE FRQSP 73/74 OPT=1

FTN 4.2+75060 01/09/78 14.02.55.

```

40 FREQ= TAN(B7./57.3)
41 FREQ= TAN(FREQ * DELT*.51
42 DELFREQ= 1.15
43 CONTINUE
44 IF ((FREQ.EQ.0..AND.DIGITL.EQ.0) GO TO 80
45 IF ((FREQ.NE.0..AND.DIGITL.EQ.0) GO TO 81
46 IF ((FREQ.EQ.0..AND.DIGITL.NE.0) GO TO 82
47 IF ((FREQ.NE.0..AND.DIGITL.NE.0) GO TO 83
48 FREQ=1.15*FREQ
49 IF (J.EQ.1) FREQ=150.
50 GO TO 84
51 IF (J.EQ.1) GO TO 85
52 FREQ=FREQ*DELFREQ
53 GO TO 84
54 FREQ=IFREQ
55 GO TO 84
56 IF (FREQ.EQ.-1) GO TO 80
57 IF (J.EQ.1) GO TO 85
58 FREQ=1.15*FREQ
59 GO TO 84
60 FREQ=TAN(FREQ * DELT*.51)
61 GO TO 84
62 IF (J.EQ.1) GO TO 85
63 FREQ=FREQ*DELFREQ
64 CONTINUE
65 PN1=CMPLX (1.0,0.0)
66 PN2=CMPLX (1.0,0.0)
67 XR= 0.0
68 XI= FREQ
69 DO 20 I=1,NN
70 IF ((FREQ.NE.-1) GO TO 21
71 XR= COS(FREQ*DELT)
72 XI= SIN(FREQ*DELT)
73 IF (I.GT.NNUM) GO TO 5
74 R1= CMPLX(XR-ROTR(I), XI-ROTI(I))
75 PN1= PN1*RN1
76 PN2= CMPLX(XR-ROOTP(I), XI-ROOTI(I))
77 RD1=PN1*RN2
78 CONTINUE
79 RD1=GAIN*RN1/RD1
80 PHI=TANG(RD1)
81 REAL1=REAL(RD1)
82 AMAG=AIMAG(RD1)
83 AMPRAT =CORT(AMAG**2+REAL1**2)
84 DR2=0.*ALOG10(AMPRAT)
85 IF (IMCO.EQ.0) GO TO 30
86 IF (IMCO.EQ.2) GO TO 40
87 SAV1(J)=DB
88 SAV2(J)=PHI
89 IF (IFORM.EQ.2) GO TO 6
90 IF (DIGITL.NF. 0 ..AND. FRPS .NE. -1) GO TO 79
91 WRITE (7,50) FREQ,DB,PHI
92 GO TO 11
93 CMFGA= ATAN(FREQ)*2./DELT
94 WRITE (3,51) FREQ,DB,CMFGA,CR,PHI
95 11 CONTINUE

```

SUBROUTINE FRQSP 73/74 CPT=1

FTN 4,2+75060

01/09/76 14.02.55.

115	TF(FCRM,FQ,0) GO TO 60	FRQSP 116
	6 WRITE(71,FREQ,09,PHI)	FRQSP 117
121	CONTINUE	FRQSP 118
	GO TO 60	FRQSP 119
40	A=0 ?-SAV1(J)	FRQSP 120
120	B=PHI?-SAV2(J)	FRQSP 121
	IF(FORM,FO,2) GO TO 7	FRQSP 122
	IF (DICTL.NE. 0 .AND. FRPS .NE. -1) GO TO 12	FRQSP 123
	WRITE(7,50) FREQ,09,PHI,A,B	FRQSP 124
	GO TO 13	FRQSP 125
125	12 WRITE(3,51) FREQ,OMEGA,09,PHI,A,B	FRQSP 126
	13 CONTINUE	FRQSP 127
	IF(FCRM,FO,0) GO TO 60	FRQSP 128
	7 WRITE(71,FREQ,09,PHI,A,B)	FRQSP 129
130	131 CONTINUE	FRQSP 130
	50 FFORMAT (F10.4,4E20.5)	FRQSP 131
	51 FORMAT (2F10.4,4E20.5)	FRQSP 132
	60 J=J+1	FRQSP 133
	IF (FREQ,LE,FFREQ) GO TO 100	FRQSP 134
135	10 CONTINUE	FRQSP 135
	IF(FCRM,FO,0) GO TO 8	FRQSP 136
	JD=99	FRQSP 137
	IF(IMOD,FO,2) WRITE(7) JD,DR,PHI,A,B	FRQSP 138
	IF(IMOD,NE,2) WRITE(7) JD,DT,PHI	FRQSP 139
140	H CONTINUE	FRQSP 140
	RETURN	FRQSP 141
140	END	FRQSP 142

```

        SUBROUTINE HESSEN (A,M,N,
     1MX,MY,MU,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        DIMENSION M(MX),A(MX,MX)
        COMMON/SUBRIT/ ISUBNAM
        IF(IISUBNAM.GE.1) WRITE(1,900)
490 FORMAT(1X,'HESSEN')
        IF (M-2) 30,30,32
320 DO 40 LC = 3,M
        N = M - LC + 3
        M1 = N - 1
        M2 = N - 2
        NI = N1
        DIV= ABS(A(N,N-1))
        DO 2 J = 1,N
        IF( ABS(A(N,J))-DIV) 2,2,1
1    NI = J
        DIV= ABS(A(N,J))
P    CONTINUE
        IF(DIV) 3,4,3
3    IF(NI = NI) 4, 7,*4
4    DO 5 J = 1,M
        DIV = A(J,NI)
        A(J,NI) = A(J,N1)
        A(J,NI) = DIV
P    CONTINUE
        DO 6 J = 1,M
        DIV = A(N1,J)
        A(N1,J) = A(N1,NI)
        A(N1,J) = DIV
A    CONTINUE
        7 DO 2K K = 1, NI
        B(K) = A(N,K)/A(N,N-1)
26    CONTINUE
        DO 45 J = 1,M
        SUM = 0.0
        IF (J = NI) 46,43,43
45    IF(B(J)) 41,43,41
41    A(N,J) = 0.0
        DO 42 K = 1,NI
        A(K,J) = A(K,J) - A(K,NI)*B(J)
        SUM = SUM + A(K,J)*B(K)
42    CONTINUE
        GO TO 46
43    DO 44 K = 1,NI
        SUM = SUM + A(K,J)*B(K)
44    CONTINUE
45    A(N1,J) = SUM
46    CONTINUE
30    RETURN
      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

SUBROUTINE INPUTV(DEL,T,U,
1 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
COMMON/CONV/READ,SYSTEM,OUTPUT,NX,NY,NU,NYC,NUC,N1,N2,DIGITL,
ISCONTIN,NUMER,FREQ,TRESP,MODEL,INSCALE,SAV,CMAT,NK2,IFLAG,
TIGO,FORM,TDT,PEAS,MIXED,MULTRT,SCALPT,ZOH,KOUNT
INTEGER READ,SYSTEM,OUTPUT,FORC,CENTUR,SAV,CMAT,READI, FREQ,TRSP
INTEGER DIGITL,SCALPT,ZOH
DIMENSION U(MX,1)

0      C   SUB-R WRITTEN SUBROUTINE CONSTRUCTING INPUT VECTOR FOR TRANSIENT
       C   RESPONSE.

       C   COMMON/SUBRWT/, ISUBNAM
       C   IF(IISUBNAM.GE.2) WRITE(1,990)
990  FORMAT(1X,"INPUTV")
       C   IF (T.TGT.0.0) RETURN
       C   READ (1,1) (U(I,J),I=1,NU)
       C   IF (EOF(1).NE.0) STOP
1   FORMAT (4F10.4)
       C   RETURN
       C   END

```

SUBROUTINE INVR

7374 OPT=1

FTN 4.2#75060

01/09/76 14.03.0F.

```

      SUBROUTINE INVR (A,N,JJJ,IT,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
C  PROGRAM AUTHORS R.E. FUNDFRIC AND R.G. EDWARDS,
C  COMPUTING TECHNOLOGY CENTER, UNION CARBIDE CORP., NUCLEAR DIV.,
C  OAK RIDGE, TENN.
C
C  CTC FOR PROGRAM NO. 9057.L
C  DIMENSION A(MX,MX),B(MX,MX)
C  COMMON/MUND/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,
C  1CCNTL9,NUM,RS,FRPS,TRESP,MODEL,NSCALE,SAV,CMAT,NK2,IFLAG,
C  1IGO,FSM,IPR,READ3,MIXED,MULTRT,SCAPLT,ZOH,KOUNT
C  INTEGER READ,SYSTEM,OUTPUT,FORM,CCNTL9,SAV,CMAT,READ3,FRPS,TRESP
C  INTEGER DIGITL,SCAPLT,ZOH
C  COMMON/SUBM/IT,ISUBNAM
C  IF(IISUBNAM.GE.2) WRITE(3,990)
C  990 FORMAT(1X,*INVR*)
C  MAT1=MX
C  MAT2=MY
C  IF (IPT.LT.1) GO TO 70
C  WRITE(*,71)
C  71 FORMAT(1X/* MATRIX ENTERING INVR AND ITS INVERSE* */
C  CALL SPIT1 (A,JJJ,JJJ,
C  1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
C  70 CONTINUE
C  IF ((JJJ.NE.1) GO TO 50
C  A(1,1)=1./A(1,1)
C  RETURN
C  71 CONTINUE
C  DO 21 I=1,JJJ
C  DO 22 J=1,JJJ
C  A(I,J)=0.
C  20 CONTINUE
C  A(I,I)=1.0
C  21 CONTINUE
C  KKEJJJ
C  NV=JJJ
C  N=1.
C  IF ((JJJ.LT.0)D=0.
C  KKH=KK-1
C  DO 9 I=1,KKM
C  D=0.0
C  DO 1 J=1,KK
C  P=A(I,J,I)
C  IF (P.LT.0) GO TO 1
C  S=0.
C  L=J
C  1 CONTINUE
C  IF (L,P0,I) GO TO 5
C  DO 2 J=1,KK
C  S=A(I,J)
C  A(I,J)=A(L,J)
C  A(L,J)=S
C  2 CONTINUE
C  IF (NV,LF,0)GO TO 4
C  DO 3 J=1,NV
C  S=A(I,J)
C  A(I,J)=A(L,J)
C  3

```

SUBROUTINE INVR

77/74 OPT=1

FTN 4.2+75060

01/09/76 14.03.08.

```

      R(I,J)=0
      3 CONTINUE
      D=-D
      IF(A(I,I).EQ.0.) GO TO 9
      IFO=I+1
      DO K=IFO,KK
      IF (A(J,I).EQ.0.) GJ TO 8
      S=A(J,I)/A(I,I)
      A(J,I)=0.0
      DO 6 K=IFO,KK
      A(J,K)=A(J,K)-A(I,K)*S
      6 CONTINUE
      IF (INV.LE.0) GO TO 3
      DO 7 K=1,NV
      R(J,K)=B(J,K)-R(I,K)*S
      7 CONTINUE
      8 CONTINUE
      9 CONTINUE
      DO 10 I=1,KK
      D=D*A(I,I)
      10 CONTINUE
      IF (INV.LE.0) GO TO 13
      KMO=KK-1
      DO 12 K=1,NV
      R(KK,K)=B(KK,K)/A(KK,KK)
      DO 12 I=1,KMO
      N=KK-I
      DO 11 J=N,KMO
      R(N,K)=R(N,K)-A(N,J+1)*B(J+1,K)
      11 CONTINUE
      R(N,K)=R(N,K)/A(N,N)
      12 CONTINUE
      13 DMAT=0.0
      IF (IDT.LT.1) GO TO 72
      CALL SPIT1 (R,JJJ,JJJ,
      1MX,MY,MU,M5,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      72 CONTINUE
      IF (IDT.EQ.0) RETURN
      DO 30 I=1,JJJ
      DO 30 J=1,JJJ
      IF (A75(R(I,J)),LT.1.E-510(I,J))=0.
      30 CONTINUE
      RETURN
      END

```

INVR 59
 INVR 60
 INVR 61
 INVR 62
 INVR 63
 INVR 64
 INVR 65
 INVR 66
 INVR 67
 INVR 68
 INVR 69
 INVR 70
 INVR 71
 INVR 72
 INVR 73
 INVR 74
 INVR 75
 INVR 76
 INVR 77
 INVR 78
 INVR 79
 INVR 80
 INVR 81
 INVR 82
 INVR 83
 INVR 84
 INVR 85
 INVR 86
 INVR 87
 INVR 88
 INVR 89
 INVR 90
 INVR 91
 INVR 92
 INVR 93
 INVR 94
 INVR 95
 INVR 96
 INVR 97
 INVR 98
 INVR 99
 INVR 100
 INVR 101
 INVR 102

THE ROUTINE LOAD 73/74 CP1=1 FTN 4.2+75061 01/09/76 14.07.39.

```

SUBROUTINE LOAD (A,B,C,H,G,F,K1,K2,K3,K4,D,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      THIS SUBROUTINE LOADS ALL MATRICES ACCORDING TO THE
      PARAMETERS, SYSTEM AND OUTPUT, USING THE SUBROUTINE LOAD1
      COMMON/CDDN/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,
1CONTUR,NUMERO,FRPS,TRESP,MODEL,NSCALE,SAV,CHAT,NK2,TFLAG,
1IGOL,FORM,IPT,READ3,MIXED,MULTRT,SCAPLT,ZDH,KOUNT
      INTEGER READ,SYSTEM,OUTPUT,FORM,CONTUR,SAV,CHAT,READ3, FRPS,TRESP
      INTEGER DIGITL, SCAPLT, ZDH
      REAL K1,K2,K3,K4
      DIMENSION A(MX,MX),B(MX,MX),C(MX,MX),H(MY,MX),G(MY,MX),F(MY,MU),
1(K1(MU,MX),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU)
      COMMON/CHWT/ISUINAH
      IF(ICHWT.EQ.2) WRITE(3,990)
      990 FORMAT(1X,*LOAD*)
      MAT1=MX
      MAT2=MX
      NMAT=0
      CALL LOAD1 (A,NX,NX,NMAT,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MU
      CALL LOAD1 (C,NX,NU,NMAT,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF (NMAT.EQ.0) GO TO 20
      MAT1=MX
      MAT2=MX
      CALL LOAD1 (G,NX,NX,NMAT,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      GO TO 40
      20 MAT2=MX
      CALL ZGT1(C,NX,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      DO 21 I=1,NX
      21 CI(I)=1.0
      40 CONTINUE
      IF (MIXED.EQ.1) GO TO 50
      GO TO (50,50,60),SYSTEM
      60 MAT1=MU
      MAT2=MX
      CALL LOAD1 (K1,NU,NX,NMAT,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF (NK2.EQ.0) GO TO 62
      CALL LOAD1 (K2,NL,NX,NMAT,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      62 CONTINUE
      IF (NK2.EQ.0) GO TO 64
      CALL LOAD1 (K3,NU,NX,NMAT,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF (NK2.EQ.0) GO TO 64
      CALL LOAD1 (K4,NU,NX,NMAT,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      64 CONTINUE
      GO TO 200
      50 MAT1=MY
      MAT2=MX
      LOAD 2
      LOAD 3
      LOAD 4
      LOAD 5
      LOAD 6
      LOAD 7
      LOAD 8
      LOAD 9
      LOAD 10
      LOAD 11
      LOAD 12
      LOAD 13
      LOAD 14
      LOAD 15
      LOAD 16
      LOAD 17
      LOAD 18
      LOAD 19
      LOAD 20
      LOAD 21
      LOAD 22
      LOAD 23
      LOAD 24
      LOAD 25
      LOAD 26
      LOAD 27
      LOAD 28
      LOAD 29
      LOAD 30
      LOAD 31
      LOAD 32
      LOAD 33
      LOAD 34
      LOAD 35
      LOAD 36
      LOAD 37
      LOAD 38
      LOAD 39
      LOAD 40
      LOAD 41
      LOAD 42
      LOAD 43
      LOAD 44
      LOAD 45
      LOAD 46
      LOAD 47
      LOAD 48
      LOAD 49
      LOAD 50
      LOAD 51
      LOAD 52
      LOAD 53
      LOAD 54
      LOAD 55
      LOAD 56
      LOAD 57
      LOAD 58

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE LOAD

73/74 OPT=1

FTN 4.2+7506I

01/09/76 14.03.39.

	CALL LOAD1 (H,NY,NX,NMAT,	LOAD	F9
60	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	LOAD	F0
	GC TO (100,5F,57,F4),OUTPUT	LOAD	F1
	55 CALL LOAD1 (G,NY,NX,NMAT,	LOAD	F2
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	LOAD	F3
	GO TO 100	LOAD	F4
65	57 MAT2=MU	LOAD	F5
	CALL LOAD1 (F,NY,NJ,NMAT,	LOAD	F6
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	LOAD	F7
	GC TO 100	LOAD	F8
	58 CALL LOAD1 (G,NY,NX,NMAT,	LOAD	F9
70	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	LOAD	F0
	MAT2=MU	LOAD	F1
	CALL LOAD1 (F,NY,NJ,NMAT,	LOAD	F2
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	LOAD	F3
	100 IF (NMIXED.EQ. 1) GO TO 200	LOAD	F4
	GO TO (200,110,2001),SYSTEM	LOAD	F5
75	110 MAT1=MU	LOAD	F6
	MAT2=MX	LOAD	F7
	CALL LOAD1 (K1,NU,NX,NMAT,	LOAD	F8
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	LOAD	F9
80	IF (NK2.EQ.0) GO TO 66	LOAD	F0
	CALL LOAD1 (K2,NU,NX,NMAT,	LOAD	F1
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	LOAD	F2
	66 CONTINUE	LOAD	F3
	MAT2=MU	LOAD	F4
	CALL LOAD1 (D,NU,NU,NMAT,	LOAD	F5
85	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	LOAD	F6
	200 RETURN	LOAD	F7
	END	LOAD	F8

OPERATING SYSTEM

73/74 CPT=1

FTN 4.2+7E060

01/09/78 14:03:48

```
      SUBROUTINE LOAD1 (A,N,M,NMAT,
     1 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      DIMENSION A(MAT1,MAT2)
      COMMON/CURRIT/ ISURNAM
      IF (ISURNAM.GE.,2) WRITE(3,990)
 990 FORMAT(1X,*LOAD1*)
      NMAT=NMAT+1
 10 FORMAT (2I10)
      READ (1,10) N1,N2
      IF (.NOT.EQ(1),NE,0) STOP
      IF (N1.EQ.N,AND,N2.EQ,M) GO TO 20
      WRITE (1,100) NMAT,N,M,NL,N2
 100 FORMAT (//,10X,* WARNING DIMENSION OF NUMBER*,I2,* MATRIX SHOULD
     1 BE *,IS*, BY*,I2,* BUT IS*,IS,* BY*,I2,//)
 15 DO 30 I=1,N
      READ (1,200) (A(I,J),J=1,M)
      IF (.NOT.EQ(1),NE,0) STOP
 30 CONTINUE
 200 FORMAT (MF10.4)
      RETURN
      END
```

SUBROUTINE MAKE

73/74 OPT=1

FTN 4.2+75061

01/09/76 14.12.48.

```
SUBROUTINE MAKE (A,B,N,M,
1MX,NY,M1,M2,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
DIMENSION A(MAT1,MAT2),B(MAT1,MAT2)
COMMON/SUMWRIT/ ISUMINAM
IF (ISUMINAM.GE.2) WRITE(1,990)
190 FORMAT(1X,*MAKE*)
DO 10 I=1,N
DO 10 J=1,M
A(I,J)=B(I,J)
10 CONTINUE
RETURN
END
```

	MAKE	2
	MAKE	3
	MAKE	4
	MAKE	5
	MAKE	6
	MAKE	7
	MAKE	8
	MAKE	9
	MAKE	10
	MAKE	11
	MAKE	12
	MAKE	13

SUBROUTINE MATRIX 73/74 CPT=1

FTN 4.2+75060

01/09/78 1+12.11.

	SUBROUTINE MATRIX (A,B,C,H,G,F,K1,K2,K3,K4,O,W1,W2,W7, 1HY,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MATE1) COMMON/COND/PEAD,SYSTEM,OUTPUT,NK,NY,NU,NKC,NUC,N1,N2,CT,SITL, 1CNTUR,NUMERS,FRPS,TREFP,MODEL,NSCALE,SAV,CMAT,NK?,IFLAG, 1TGC,FORM,IPT,READ3,MIXED,MULTRT,SCAPLT,ZOH,KOUNT INTEGER READ,SYSTEM,OUTPUT,FCRM,CNTUR,SAV,CMAT,FRPS,TREFP,READ3 INTEGER DIGIT, SCAPLT, ZOH COMMON/ACOND/ CBLT,FINALT,TFREQ,DELFFQ,GATH1,GAIN2,H REAL K1,K2,K3,K4,IFREQ,H DIMENSION A(MY,MX),B(MY,MU),C(MY,MX),H(MY,MX),G(MY,MX),F(MY,MU), 1K1(MU,MX),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU), 2W1(MX,MX),W2(MY,MX),W3(MY,MX) DIMENSION GRAPH(20,5),BLOCK(20,3),NUMER(20,5),DENOM(20,5), YGAIN(20),STATE(20,4),ITHINY(30),ITHINU(20),YTUV(20,2), Y_ZTCU(20,2),NXYU(3),YZTOK(20,2) REAL NUMER INTEGER GRAPH,BLOCK,STATE,YTUV,ZTCU,YZTOK COMMON /RALKDATA/ NUMER,DENOM,GAIN,GRAPH,BLOCK,STATE,YTUV, Y_ZTCU,Y_ZTOK,ITHINY,ITHINU,NALCK,NYTCU,NZTCU,NXYU,NZTOK, Y_NXT,NYT,NUT,NY1,HU1 C C USR WRITTEN SUBROUTINE TO CONSTRUCT SYSTEM MATRICES UNDER CONTROL C OF CONDITION CODES C COMMON/SUBWIT/ ISUBNAM IFI(ISUPNAM.GE.2) WRITE(3,990) 990 FORMAT(1X,*MATRIX SUB 2*) RETURN END	MATRIX 2 MATRIX 3 MATRIX 4 MATRIX 5 MATRIX 6 MATRIX 7 MATRIX 8 MATRIX 9 MATRIX 10 MATRIX 11 MATRIX 12 MATRIX 13 MATRIX 14 MATRIX 15 MATRIX 16 MATRIX 17 MATRIX 18 MATRIX 19 MATRIX 20 MATRIX 21 MATRIX 22 MATRIX 23 MATRIX 24 MATRIX 25 MATRIX 26 MATRIX 27 MATRIX 28 MATRIX 29 MATRIX 30
--	--	---

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE MULT

73/74 OPT=1

FTN 4.2+75060

01/09/75 14.12.53.

```
SUBROUTINE MULT (A,B,C,N,M,K,
1 M1,M2,M3,M4,M5,M6)
2 DIMENSION A(MAT1,MAT2),B(MAT3,MAT4),C(MAT5,MAT6)
3 COMMON/SUBRWT/ TSUTNAM
4 IF(TSUTNAM.GE.2) WRITE(3,990)
5 990 FORMAT(1X,*MULT*)
6 DO 10 I=1,N
7 DO 10 L=1,K
8 XX=0.0
9 DO 11 J=1,M
10 XX=XX+A(I,J)*B(J,L)
11 CONTINUE
12 C(I,L)=XX
13 CONTINUE
14 RETURN
15 END
16 MULT 2
17 MULT 3
18 MULT 4
19 MULT 5
20 MULT 6
21 MULT 7
22 MULT 8
23 MULT 9
24 MULT 10
25 MULT 11
26 MULT 12
27 MULT 13
28 MULT 14
29 MULT 15
30 MULT 16
31 MULT 17
```

```

      SUBROUTINE NMRATR (NN,A,B,C,D,E,F,G,H,I,J,K,OCTR,ROOTI,ROTR,ROT1,Z,V,
     141,W,IWT,SAV1,SAV2,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
C
C      THIS SUBROUTINE DETERMINES THE NUMERATORS OF TRANSFER
C      FUNCTIONS BY FINDING THE M-MATRIX WHOSE EIGENVALUES ARE
C      THE 1-D EIGENVALUES OF ROT1. SUBROUTINE EIGEN IS CALLED TO FIND
C      THE 1-D EIGENVALUES.
C
C      CODE ORIGINATED BY G. NORIS JULY 5 73
C      V AND ROTR, AND Z AND ROT1 ARE SAME MATRIX
C
C      COMMON/COND/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,
C      TCONTUR,NUMER1,TREFS,TREFS0,MODEL,NSCALC,SAV,CMAT,NK2,IFLAG,
C      TIGD,FORM,IP1,PHAD3,MIXED,MULTAT,SCAPLT,ZOH,KOUNT
C      INTEGER,REAL,SYSTEM,OUTPUT,FORM,CONTUR,SAV,CMAT,READ3,
C      1,READ4,TREFS
C      INT-GEN DIGITL,SCAPLT,ZOH
C      COMMON/ACOND/DELT,FINALT,TREFC,TREFD,DELFRD,GAIN1,GAIN2,MN
C      REAL,INPT(10),OUTPT(20),TITLE(8)
C      COMMON/LABEL/INPT,OUTPT,TITLE
C      REAL,IFP,Q,MU
C      DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),D(MY,MX),E(MY,MU),
C      10(MU,MU),
C      H1(MX,MX),H2(MY,MX),H3(MY,MX),ROCTR(MX),ROOTT(MY),ROTR(MX),
C      10(MY),Z(MX),V(MX),SAV1(MS),SAV2(MS)
C      DIMENSION A(15,15),B(15,15),C(15,15),D(15,15),E(15,15),
C      1,F(15,10),G(10,10)
C      DIMENSION H1(15,15),H2(15,15),H3(15,15)
C      DIMENSION ROCTR(15),ROOTT(15),ROTR(15),ROT1(15)
C      DIMENSION SAV1(200),SAV2(200)
C      DIMENSION V(15)
C      COMMON/SUMPIT/ TSUMHAM
C      INTEGER,ON
C      DATA ONE/1,DUMY/0.0/
C      IF(TSUMHAM.GE.1) WRITE(3,940)
C 900 FORMAT(1X,*NMRATR*)
C      DO 100 I=1,NU
C      IPONE=0
C      DO 100 J=1,NY
C      NN=0
C      DO 500 IX=1,NX
C      IF (H(I,IX).NE.0.0) GO TO 501
C 500 CONTINUE
C      WRITE(1,502)
C 502 FORMAT(1X,*ROW #,IP#, OF H MATRIX IS NULL *//)
C      WRITE(1)ONE,DUMY,DUMY
C      GO TO 100
C 100 CONTINUE
C      IF (E(I,JJ).EQ.0.0) GO TO 410
C      NN=MNN
C      GO TO 431
C 410 DO 420 K=1,NN
C      ROTR(K)=0,
C      IF (H(I,K).NE.0.0)ROTR(K)=1.
C 420 CONTINUE
C      DO 430 LL=1,NN
C      GO 421 K=1,NN
C
      NMRATR   2
      NMRATR   3
      NMRATR   4
      NMRATR   5
      NMRATR   6
      NMRATR   7
      NMRATR   8
      NMRATR   9
      NMRATR  10
      NMRATR  11
      NMRATR  12
      NMRATR  13
      NMRATR  14
      NMRATR  15
      NMRATR  16
      NMRATR  17
      NMRATR  18
      NMRATR  19
      NMRATR  20
      NMRATR  21
      NMRATR  22
      NMRATR  23
      NMRATR  24
      NMRATR  25
      NMRATR  26
      NMRATR  27
      NMRATR  28
      NMRATR  29
      NMRATR  30
      NMRATR  31
      NMRATR  32
      NMRATR  33
      NMRATR  34
      NMRATR  35
      NMRATR  36
      NMRATR  37
      NMRATR  38
      NMRATR  39
      NMRATR  40
      NMRATR  41
      NMRATR  42
      NMRATR  43
      NMRATR  44
      NMRATR  45
      NMRATR  46
      NMRATR  47
      NMRATR  48
      NMRATR  49
      NMRATR  50
      NMRATR  51
      NMRATR  52
      NMRATR  53
      NMRATR  54
      NMRATR  55
      NMRATR  56
      NMRATR  57
      NMRATR  58

```

SUBROUTINE NMRATR 73/74 CPT=1 FTN 4.2+75060 01/09/76 14.12.5F.

```

      IF (PCTP(K) .EQ. 0.0 .OR. P(K,JJ) .EQ. 0.0) GO TO 421
      NAUM=NN-1L
      GO TO 421
 421  CONTINUE
      ITEST=0
      DO 422 K=1,NN
      ROTI(K)=0.
 422  CONTINUE
      DO 422 K=1,NN
      DO 422 L=1,NN
      IF (K.EQ.L) GO TO 422
      IF (POTR(K).EQ.0.0 .OR. A(K,L).EQ.0.0) GO TO 422
      POTI(L)=1.
      ITEST=1
 422  CONTINUE
      IF (ITEST.EQ.0) GO TO 440
      DO 423 K=1,NN
      ROTR(K)=ROTI(K)
 423  CONTINUE
 430  CONTINUE
 440  WRITE (3,441) JJ,I
 441  FORMAT (/10X,*NUMBER*,I4,* INPUT DOES NOT EXCITE NUMBER*,I4,* 0U
      1TPUT*)
      GO TO 300
 451  W2(1,1)=F(I,JJ)
      MM=N+1
      DO 452 K=2,MM
      W2(1,K)=-W1(I,K-1)
      W2(K,1)=W(K-1,JJ)
      DO 453 L=2,MM
      W2(K,L)=-A(K-1,L-1)
 452  CONTINUE
 453  CONTINUE
      IF (IPT.LT.2) GO TO 240
      WRITE (3,241)
 241  FORMAT (/* NUMERATOR MATRICES */)
      MAT1=4X
      MAT2=4X
      CALL SPITI (W2,MM,MM,
      1MY,2MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 240  CONTINUE
      IT=1
      IF (W2(1,1)) 181.8.181
      4 DO 180 II=1,LL
      DO 180 L<2,MM
      IF (W2(K,1).NE.0.0) GO TO 22
 10  CONTINUE
      GO TO 200
 20  M=K
      Z(II)=W2(M,1)
      DO 20 K=1,N
      DO 20 L=1,N
      W3(K,L)=-(W2(K+1,1)/Z(II))*(W2(1,L+1)+W2(M,L+1))
 10  CONTINUE
 20  CONTINUE
      DO 30 K=1,N
      DO 30 L=1,N

```

NMRATR 59
 NMRATR 60
 NMRATR 61
 NMRATR 62
 NMRATR 63
 NMRATR 64
 NMRATR 65
 NMRATR 66
 NMRATR 67
 NMRATR 68
 NMRATR 69
 NMRATR 70
 NMRATR 71
 NMRATR 72
 NMRATR 73
 NMRATR 74
 NMRATR 75
 NMRATR 76
 NMRATR 77
 NMRATR 78
 NMRATR 79
 NMRATR 80
 NMRATP 81
 NMRATR 82
 NMRATR 83
 NMRATR 84
 NMRATR 85
 NMRATR 86
 NMRATR 87
 NMRATR 88
 NMRATR 89
 NMRATR 90
 NMRATP 91
 NMRATR 92
 NMRATR 93
 NMRATR 94
 NMRATR 95
 NMRATR 96
 NMRATR 97
 NMRATP 98
 NMRATP 99
 NMRATR 100
 NMRATR 101
 NMRATR 102
 NMRATR 103
 NMRATR 104
 NMRATR 105
 NMRATP 106
 NMRATR 107
 NMRATR 108
 NMRATR 109
 NMRATR 110
 NMRATR 111
 NMRATR 112
 NMRATR 113
 NMRATR 114
 NMRATR 115

```

115      W1(K,L)=W3(K,L)+W2(K+1,L+1)
20  CONTINUE
30  CONTINUE
40  IF (M=2) 32,42,32
50  DO 60 K=1,N
60  STORE=W3(1,K)
70  W3(1,K)=W3(M-1,K)
80  W3(M-1,K)=STORE
90  CONTINUE
100  DO 50 K=1,N
110  G1=0.0
120  DO 45 L=1,N
130  G1=W3(K,L)*(W2(L+1,1)/Z(L))+G1
140  CONTINUE
150  W3(K,M-1)=G1
160  CONTINUE
170  IF (M=2) 52,62,52
180  DO 60 K=1,N
190  STORE=W3(K,1)
200  W3(K,1)=W3(K,M-1)
210  W3(K,M-1)=STORE
220  CONTINUE
230  G1=0.0
240  DO 71 KKK=1,N
250  G1=W2(1,KKK+1)*W2(KKK+1,1)+G1
260  CONTINUE
270  DO 70 K=1,N
280  DC 70 L=1,N
290  W2(K,L)=W3(K,L)
300  CONTINUE
310  W2(1,1)=G1/Z(1)
320  N=M-1
330  M=M+1
340  IF (OPT.LT.2) GO TO 180
350  CALL SPITS (M2,M4,MH,
360  1MH,1V,1U,1S,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
370  CONTINUE
380  CONTINUE
390  FOLLOWING STATEMENT ADDED SEPT 1972 DUE TO CDC IBM DO LOOP
400  INDEX DIFFERENCE
410  IF (II.LT.0) II=II+1
420  IN THE FORM---
430  DO 180 II=1,LL WHERE LL=1
440  II TERMINAL VALUE IS 2 ON CDC
450  II TERMINAL VALUE IS 1 ON IBM
460  IF (N.EQ.0) GO TO 197
470  DO 190 K=1,N
480  DO 185 L=1,N
490  W3(K,L)=-W2(K+1,L+1)+ ((W2(K+1,1)*W2(1,L+1))/W2(1,1))
500  CONTINUE
510  CONTINUE
520  IF (II) 193,195,193
530  DO 195 K=1,II
540  DE=DE*Z(KJ)
550  CONTINUE
560  CONTINUE
570  CONTINUE

```

```

      IF (N,NE,0) GO TO 130
      WRITE (3,210) OUTPT(II), INPT(JJJ),DE
      210 FORMAT (//10X,*THE*,A10,*/*,A10,*NUMERATOR GAIN IS *,E12.4,
      *THESE ARE NO ZEROS*)/
      GO TO 140
      140 DC 400 K=1,N
      DO 400 J=1,N
      H1(K,J)=H3(K,J)
      400 CONTINUE
      IF (IPT,LT,0) GO TO 243
      CALL SPIT1(H1,N,,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      243 CONTINUE
      WRITE (3,209) OUTPT(II), INPT(JJJ),DE
      209 FORMAT (//10X,*THE *,A10,*/*10,* NUMERATOR GAIN IS*,E12.4//*
      * 1 ZEROS OF THE TRANSFER FUNCTION ARE//20X,*REAL PART*,15X,*IMAGIN
      ?ARY PART*/)
      IF (MN,NE,0,AND. SYSTEM ,NE, 3) WRITE (3,1000)
      1000 FORMAT (/* FOR MODIFIED Z-TRANSFERS, ADD Z=0, POLE*/)
      CALL FIGEN (N,H1,W2,W3,ROTR,ROTI,Z,V,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL CPMT (Z,ROTR,ROTI,N,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      WRITE (3,821)
      821 FORMAT (//10X,* THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ARE
      *RED FRM THE LOWEST POWER OF S//)
      NX1=N+1
      WRITE (3,83) (Z(IA), IA=1,NX1)
      83 FORMAT (E20.8)
      198 CONTINUE
      IF (IDIGITL,EQ,0) GO TO 110
      IF (MN ,EQ, 0) GO TO 113
      NSAVE=N
      205 N=N+1
      NX=NX+1
      IF (I ,GT, 1 ,OR, JJ ,GT, 1) CC TC 113
      RCOTR(NX)= 0.0
      ROOTT(NX)= 0.0
      210 CONTINUE
      IF (IDIGITL ,EQ, 0 ,OR, SYSTEM ,EQ, 3 ,OR, FRPS ,EQ, -1) GO TO 110
      IF (I ,GT, 1 ,OR, JJ ,GT, 1) CC TO 110
      CALL ZTOW (N,N,DE,2,Z,ROTR,RCOTR,ROTR,ROTI,L1,DEE,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      WRITE (3,111)
      111 FORMAT (//10X,* THE N-PLANE POLES ARE //)
      WRITE (3,112) (RCOTR(K),ROOTT(K),K=1,N)
      112 FORMAT (/(2E30.8))
      110 CONTINUE
      IF (IDIGITL,EQ,0,OR,SYSTEM,EQ,3,OR,FRPS,EQ,-1) GO TO 120
      CALL ZTOW (N,N,DE,2,Z,ROTR,RCOTR,ROTR,ROTI,L1,DEE,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      WRITE (3,121) CE
      121 FORMAT (//10X,* THE N-PLANE TRANSFER FUNCTION GAIN IS *,E12.4/*
      * THE ZEROS ARE*)
      122 WRITE (3,122) (ROTR(K),ROTI(K),K=1,N)
      120 CONTINUE
      IF (FRPS,EQ,-1) GO TO 301
      
```

SUBROUTINE NMRATR 73/74 OPT=1

FTN 4.2+75050 01/09/76 14.12.5t.

```

230      IFR=FRPS+1
          GO TO (311,310,320) IFR
310      IF (MODL.EQ.0) GO TO 301
          IF (MOD(I,2)).EQ.1) GO TO 302
          IF (IMOD.EQ.2) GO TO 302
          IMOD=2
301      GO TO 301
302      IMOD=1
303      CALL FRORSP (N,NN,DE,IMOD,ROOTR,ROTI,ROTR,ROTI,SAV1,SAV2,
              1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
          GO TO 311
304      CALL PSD (N,NN,DE,ROOTR,ROTI,ROTR,ROTI,SAV1,SAV2,I,JJ,
              1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
          GO TO 311
305      WRITE (3,211) JJ
311      FORMAT (//10X,*COLUMN *,I2,* CF THE B MATRIX IS NULL*)
312      IF (MN.EQ.0) GO TO 300
          NX=NSAV
          NN=NSAV
300      CONTINUE
          END

```

230	NMRATR	230
	NMRATR	231
	NMRATR	232
	NMRATR	233
	NMRATR	234
	NMRATR	235
	NMRATR	236
	NMRATR	237
	NMRATR	238
	NMRATR	239
	NMRATR	240
	NMRATR	241
	NMRATR	242
	NMRATR	243
	NMRATR	244
	NMRATR	245
	NMRATR	246
	NMRATR	247
	NMRATR	248
	NMRATR	249
	NMRATR	250
	NMRATR	251

```

SUBROUTINE PSD (INNUM,NN,GAIN,ROOTR,ROTI,ROTR,ROTI,SAV1,SAV2,
1INY,INU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)          PSD   2
                                                       PSD   3
                                                       PSD   4
                                                       PSD   5
                                                       PSD   6
                                                       PSD   7
                                                       PSD   8
                                                       PSD   9
                                                       PSD  10
                                                       PSD  11
                                                       PSD  12
                                                       PSD  13
                                                       PSD  14
                                                       PSD  15
                                                       PSD  16
                                                       PSD  17
                                                       PSD  18
                                                       PSD  19
                                                       PSD  20
                                                       PSD  21
                                                       PSD  22
                                                       PSD  23
                                                       PSD  24
                                                       PSD  25
                                                       PSD  26
                                                       PSD  27
                                                       PSD  28
                                                       PSD  29
                                                       PSD  30
                                                       PSD  31
                                                       PSD  32
                                                       PSD  33
                                                       PSD  34
                                                       PSD  35
                                                       PSD  36
                                                       PSD  37
                                                       PSD  38
                                                       PSD  39
                                                       PSD  40
                                                       PSD  41
                                                       PSD  42
                                                       PSD  43
                                                       PSD  44
                                                       PSD  45
                                                       PSD  46
                                                       PSD  47
                                                       PSD  48
                                                       PSD  49
                                                       PSD  50
                                                       PSD  51
                                                       PSD  52
                                                       PSD  53
                                                       PSD  54
                                                       PSD  55
                                                       PSD  56
                                                       PSD  57
                                                       PSD  58

```

C THIS SUBROUTINE COMPUTES THE POWER SPECTRUM CORRESPONDING
C TO THE TRANSFER FUNCTION WHOSE POLES ARE (ROCTR,ROTI) AND
C WHOSE ZEROS ARE (ROTR,ROTI). IT IS ASSUMED THAT THE INPUT
C TO THE TRANSFER FUNCTION IS A WHITE NOISE PROCESS OF UNITY
C VARIANCE AND HENCE THE PSD IS GIVEN BY THE SQUARE OF THE
C MODULUS OF THE TRANSFER FUNCTION. THIS ANY CORRELATION
C DESIRED IN THE RANDOM PROCESS EXCITING THE SYSTEM SHOULD
C BE INCLUDED AS A SHAPING FILTER IN THE "A" MATRIX WHICH
C IS EXCITED BY WHITE NOISE. NOTE THAT ONLY EXPONENTIALLY
C CORRELATED PROCESSES MAY BE SO STRUCTURED.

C

DIMENSION ROOTP(MX),ROTR(MX),ROT(I,MX),SAV1(MS),SAV2(MS)

COMPLEX RN1,RN2,PDL,RD2

COMMON/COND/READ,SYSTEM,OUTPUT,NY,NY,NU,NXC,NUC,N1,N2,DIGITL,

1CONTUR,NUMERS,FRPS,TRESP,MODEL,NSCALE,SAV,CHAT,NK2,IFLAG,

?IGC,FOPM,IPR,READY,4IXED,MULTPT,SCAPLT,ZOH,KOUNT

INTEGER READ,SYSTEM,OUTPUT,FORM,1CONTUR,SAV,CHAT,READ3,FRPS,TRESP

COMMON/ACOND/ DELT,FINALT,IFREQ,FFREQ,DELFRQ,GAIN1,GAIN2,M

INTEGER DIGITL,SCAPLT,ZOH

REAL IFRQ,4

REAL PST

COMMON/LABEL/INPT,OUTPT,TITLE

REAL INPT(10), OUTPT(20),TITLE(8)

COMMON/SUBRIT/TSUBRIT

DATA PST/10HSPEC /

IF(TURNAM.GE.2) WRITE(3,990)

990 FORMAT(1X,*PSD*)

IF(FGRM,GT,0) WRITE(7) PST,TITLE,SYSTEM,MODEL,DIGITL,SCAPLT

VAP=*.0

X=0.0

J=1

IF(FORM,GT,0) WRITE(7) OUTPT(INY),INPT(INU)

WRITE(7,60) OUTPT(I(NY)),IMP(I(NU))

60 FORMAT(1/10X,A10,/*A10,* POWER SPECTRAL DENSITY*/10X,*FREQUENCY,R

1AD/SEC*4X,*PSD MAGNITUDE*)

FFRD=IFREQ

IF (IFRD,NE,0.) GO TO 70

FREQ=1.

FFRD=150.

70 CONTINUE

L00 CONTINUE

IF (IFRD,NE,0.) GO TO 71

FREQ=1.15*FREQ

GO TO 72

71 FREQ=DELFRQ*FREQ

72 CONTINUE

RN1=CMPLX (1.0,0.0)

RC1=CMPLX (1.0,0.0)

DO 70 I=1,NN

IF (I,GT,NNUM) GO TO 5

RN2=CMPLX (-ROTR(I),FREQ -ROTI(I))

PN1=RN1*RN2

PN2=CMPLX (-ROTR(I),-FREQ -ROTI(I))

SUBROUTINE R10 11/17/74 CR7=1 F7N 4.2+75461 01/19/76 14.13.07.

 60 P00 59
 P00=PN1*PN2 P50 60
 P01=RD1*RD2 P50 61
 P02=COMPLEX (-ROOTR(I), -FREQ -ROOTI(I)) P50 62
 R01=R01*RD2 P50 63
 R01=RN1*RN2 P50 64
 20 CONTINUE P50 65
 SAV1(J)=GAIN**2*PN1/RD1 P50 66
 VAR=VAR+SAV1(J)*FREQ P50 67
 IF(FORM, EQ, 2) GO TO 6 P50 68
 WRITE(1,501) FREQ ,SAV1(J) P50 69
 IF(FORM, EQ, 0) GO TO 40 P50 70
 WRITE(7) FREQ ,SAV1(J) P50 71
 40 CONTINUE P50 72
 50 FORMAT(14X,F8.4,E12.4) P50 73
 X=FREQ P50 74
 J=J+1 P50 75
 IF (FREQ, LE, FFREQ) GO TO 100 P50 76
 10 CONTINUE P50 77
 WRITE(3,301) VAR P50 78
 30 FORMAT(10X,*THE VARIANCE IS*E12.4)
 JD=99 P50 79
 WRITE(7) JD, SAV1(I) P50 80
 RETURN P50 81
 END P50 82

ORIGINAL PAGE IS
 OF POOR QUALITY

SUBROUTINE QREIG 73/74 OPT=1

FTN 4.2+75.060

01/09/76 14.13.10.

```

SUBROUTINE QREIG (M,ROOTR,ROOTI,IFRNNT,A,
1 MX,MY,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
2 DIMENSION A(MX,MY),ROOTR(MX),FCOTI(MX)
3 COMMON/SUBRIT/ TSUBNAME
4 IF(IISUBNAME.GE.2) WRITE(3,990)
5 FORMAT(1X,*QREIG*)
6 N = M
7 IF(IFRNNT) AC,81+80
80 WRIT(3,104)
9 A1 ZERO = 0.0
10 JJ=1
117 XNN=0.0
12 XNP=0.0
13 AA = 0.0
14 R = 0.0
15 C = 1.0
16 DD = 0.0
17 R=C.0
18 SIG=0.0
19 ITER = 0
20 IF(N>2) 13,14,17
21 IF(IFRNNT) A2,83,82
22 WRIT(3,105)A(1,1)
23 PCOTI(1) = A(1,1)
24 ROOTI(1) = 0.0
25 CONTINUE
26 RETURN
27
28 JJ=-1
29 X = (A(N-1,N-1) - A(N,N))*2
30 S = 4.0*A(N,N-1)*A(N-1,N)
31 ITER = ITER + 1
32 IF (X.EQ.0.0.DR. ABS(S/X) .GT. 1.0E-8) GO TO 15
33 IF(X.EC.0.0)GC TO 15
34 IF(ABS(S/X).GT.1.0E-8) GO TO 15
35 IF ( ABS(A(N-1,N-1))- ABS(A(N,N)) ) 32,32,31
36 E = A(N-1,N-1)
37 G = A(N,N)
38 GO TO 33
39 G = A(N-1,N-1)
40 E = A(N,N)
41 F = 1.
42 H = 1.
43 GO TO 24
44 S = X + S
45 Y = A(N-1,N-1) + A(N,N)
46 IF(S) 18,19,19
47 S0= SORT(S)
48 F=0.0
49 H=0.0
50 IF (X) 21,21,22
51 E=(X-S0)/2.0
52 G=(X+S0)/2.0
53 GC TO 24
54 G=(X-S0)/2.0
55 E=(X+S0)/2.0
56 GC TO 24
57 F= SORT (-S)/2.0
58

```

SUBROUTINE DREFIG 73774 OPT=1 FTN 4.2+7F.160 01/19/71 14.13.17.

 10 T=EX/2,0
 11 F=0
 12 IF(1.0E-10*(APS(G)+F))
 13 IF((ABS(A(N-1,N-2)).GT. E) GO TO 26
 14 IF(IFERNT).NE.85,84
 15 85 WRITE (3,105)E,F,ITER
 16 WRITE (3,105)G,H
 17 IF(POOT(N)=E)
 18 POOT(N)=F
 19 POOT(N-1)=G
 20 POOT(N-1)=H
 21 N=N-2
 22 IF(JJ).LT.177,177
 23 IF(ABS(A(N,N-1)).GT. 1.0E-10*ABS(A(N,N))) GO TO 50
 24 IF(IFERNT).NE.87,86
 25 86 WRITE (3,105)A(N,N),ZERO,ITER
 26 POOT(N)=A(N,N)
 27 POOT(N)=1.0
 28 N=N-1
 29 GO TO 127
 30 GO IF(ABS(XNN/A(N,N-1))-1.0)-1.0E-6) .GE.63,63,E2
 31 IF(ABS(XN2/A(N-1,N-2))-1.0)-1.0E-6) .GE.63,63,700
 32 60= ABS(A(N,N-1))- ABS(A(N-1,N-2))
 33 IF ((ITER=15) 53,164,166
 164 IF(VD).LT.5,165,166
 165 P=A(N-1,N-2)**2
 SIG=2.1*A(N-1,N-2)
 166 GO TO 167
 167 P=A(N,N-1)**2
 SIG=2.0*A(N,N-1)
 168 GO TO 169
 169 IF(VD).LT.67,67,68
 68 IF(IFRNT).NE.85,88
 89 WRITE (3,107)A(N-1,N-2)
 GO TO 84
 87 IF(IFRNT).NE.87,89
 89 WRITE (3,107)A(N,N-1)
 GO TO 86
 700 IF(IFERNT.GT. 50) GO TO 53
 701 IF(IFERNT.GT. 5) GO TO 53
 702 IF(E).NE.E+F+F
 703 IF (E) 702,601,702
 704 GT=G*G+H*H
 705 IF (GT) 703,601,703
 706 CONTINUE
 71= ((E-AA)**2+(F-B)**2)/(E*E+F*F)
 72= ((G-C)**2+(H-D)**2)/(G*G+H*H)
 73 IF(71>0.25) 51,F1,52
 51 IF(72>0.25) 53,53,54
 53 D=F*G-F*H
 SIG=E+F
 54 GO TO 60
 55 D=F*E
 SIG=E+F
 GO TO 60
 1REIG 60
 1REIG 61
 1REIG 62
 1REIG 63
 1REIG 64
 1REIG 65
 1REIG 66
 1REIG 67
 1REIG 68
 1REIG 69
 1REIG 70
 1REIG 71
 1REIG 72
 1REIG 73
 1REIG 74
 1REIG 75
 1REIG 76
 1REIG 77
 1REIG 78
 1REIG 79
 1REIG 80
 1REIG 81
 1REIG 82
 1REIG 83
 1REIG 84
 1REIG 85
 1REIG 86
 1REIG 87
 1REIG 88
 1REIG 89
 1REIG 90
 1REIG 91
 1REIG 92
 1REIG 93
 1REIG 94
 1REIG 95
 1REIG 96
 1REIG 97
 1REIG 98
 1REIG 99
 1REIG 100
 1REIG 101
 1REIG 102
 1REIG 103
 1REIG 104
 1REIG 105
 1REIG 106
 1REIG 107
 1REIG 108
 1REIG 109
 1REIG 110
 1REIG 111
 1REIG 112
 1REIG 113
 1REIG 114
 1REIG 115

SUBROUTINE DPTI

7/17/74 09:15:11

FTN 4.2#75060

01/09/76 14:13:10.

115	IF (Z=0.0E0) AS,SG,SD,SDT	QREIG 116
	SD=0.0E0	2REIG 117
	SDT=0.0E0	QREIG 118
	SD=0.0E0	2REIG 119
F01	R = -1.0	QREIG 120
	DT = -1.0	2REIG 121
129	SD(XX1:N,N-1)	QREIG 122
	XX2=A(N-1,N-2)	2REIG 123
	CALL DST (N,2,SIG,0,A,	QREIG 124
	1 MX,MY,MU,MG,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)	QREIG 125
128	A&S	2REIG 126
	REF	QREIG 127
	C=G	2REIG 128
	DDH	QREIG 129
	GC TO 12	2REIG 130
130	104 FORMAT (//,1X, 9HREAL PART, 6X, 14HIMAGINARY PART, 2X	QREIG 131
	1 13HTAK N AD ZERO 6X 4HTPR //)	QREIG 132
105	FORMAT (1X,2E15.8,3X,E15.8, 42X [3])	QREIG 133
107	FORMAT (74X, E15.8)	2REIG 134
	END	QREIG 135

```

SUBROUTINE ORT      74774   OPT=1           FTN 4.2+75060   01/09/72  14.13.16

        SUBROUTINE ORT (IN,R,SIG,D,A,
     1MX,MY,MU,M7,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        DIMENSION A(MX,MY),PSI(12),G(3)
        COMMON/CUBRWT/ ISUBNAM
        IF (ISUBNAM.GE.2) WRITE(3,990)
990    FORMAT(1Y,*ORT*)
        N1 = N - 1
        IA = N - 2
        IF = IA
        IF(N-3) 101,10,60
60      DO 12 J = 3,N1
         J1 = N - J
         IF( ABS(A(J1+1,J1+1))-0) 10,10,11
11      DEN = A(J1+1,J1+1)*(A(J1+1,J1+1)-SIG)+A(J1+1,J1+2)*A(J1+2,J1+1)+P
         IF(D,N)=A(J1+1,J1+2)
61      IF( ABS(A(J1+1,J1+2))*A(J1+2,J1+1)*( ABS(A(J1+1,J1+1)+A(J1+2,J1+2)
         -SIG)+ABS(A(J1+1,J1+2))-0) 10,10,12
12      IF=J1
13      DO 14 J=1,TP
         J1=IF-J+1
         IF( A( P(A(J1+1,J1))-0) 13,13,14
14      IF=J1
15      DO 16 IP=1,TP,N1
         IF(I-IP)=16,15,16
16      G(1)=A(IP,IP)*(A(IP,TP)-SIG)+A(IP,IP+1)*A(IP+1,IP)+P
         G(2)=A(IP+1,IP)*(A(IF,TP)+A(IP+1,IP+1)-SIG)
         G(3)=A(IP+1,IP)*A(IP+2,IP+1)
         A(IP+2,IP)=0.0
         GO TO 13
17      G(1)=A(I,I-1)
         G(2)=A(I+1,I-1)
         IF(I-IA)=17,17,18
18      G(1)=A(I+2,I-1)
         GO TO 19
19      G(1)=0.0
19      XK= - SIG*SQRT(G(1)**2+G(2)**2+G(3)**2), G(1))
20      IF(YK)=23,24,23
21      AL=G(1)/K**1.0
22      PSI(1)=G(2)/(G(1)+XK)
23      PSI(2)=G(3)/(G(1)+XK)
         GO TO 26
24      AL=2.0
         PSI(1)=0.0
         PSI(2)=0.0
25      IF(I-IQ)=26,27,26
26      IF(I-IP)=29,28,29
27      A(I,I-1)=-A(I,I-1)
         GO TO 27
28      A(I,I-1)=-XX
29      DO 30 J=I,N
         IF(I-IA)=31,31,32
30      G=PSI(2)*A(I+2,J)
         GO TO 33
31      G=0.0
32      G=AL*(A(I,J)+PSI(1)*A(I+1,J)+C)
         A(I,J)=A(I,J)-F
         A(I+1,J)=A(I+1,J)-PSI(1)*E
33      IF(I-IP)=34,34,35
         GO TO 36
34      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 37
35      G=0.0
36      IF(I-IP)=38,38,39
         GO TO 39
37      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 40
38      G=0.0
39      IF(I-IP)=41,41,42
         GO TO 42
40      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 43
41      G=0.0
42      IF(I-IP)=44,44,45
         GO TO 45
43      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 46
44      G=0.0
45      IF(I-IP)=47,47,48
         GO TO 48
46      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 49
47      G=0.0
48      IF(I-IP)=50,50,51
         GO TO 51
49      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 52
50      G=0.0
51      IF(I-IP)=53,53,54
         GO TO 54
52      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 55
53      G=0.0
54      IF(I-IP)=56,56,57
         GO TO 57
55      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 58
56      G=0.0
57      IF(I-IP)=59,59,60
         GO TO 60
58      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 61
60      G=0.0
61      IF(I-IP)=63,63,64
         GO TO 64
62      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 65
63      G=0.0
64      IF(I-IP)=67,67,68
         GO TO 68
65      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 69
66      G=0.0
67      IF(I-IP)=70,70,71
         GO TO 71
68      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 72
69      G=0.0
70      IF(I-IP)=73,73,74
         GO TO 74
71      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 75
72      G=0.0
73      IF(I-IP)=77,77,78
         GO TO 78
74      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 79
75      G=0.0
76      IF(I-IP)=80,80,81
         GO TO 81
77      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 82
78      G=0.0
79      IF(I-IP)=84,84,85
         GO TO 85
80      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 86
81      G=0.0
82      IF(I-IP)=88,88,89
         GO TO 89
83      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 90
84      G=0.0
85      IF(I-IP)=92,92,93
         GO TO 93
86      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 94
87      G=0.0
88      IF(I-IP)=96,96,97
         GO TO 97
89      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 98
90      G=0.0
91      IF(I-IP)=100,100,101
         GO TO 101
92      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 102
93      G=0.0
94      IF(I-IP)=104,104,105
         GO TO 105
95      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 106
96      G=0.0
97      IF(I-IP)=108,108,109
         GO TO 109
98      G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 110
99      G=0.0
100     IF(I-IP)=113,113,114
         GO TO 114
101     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 115
102     G=0.0
103     IF(I-IP)=117,117,118
         GO TO 118
104     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 119
105     G=0.0
106     IF(I-IP)=122,122,123
         GO TO 123
107     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 124
108     G=0.0
109     IF(I-IP)=127,127,128
         GO TO 128
110     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 129
111     G=0.0
112     IF(I-IP)=132,132,133
         GO TO 133
113     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 134
114     G=0.0
115     IF(I-IP)=137,137,138
         GO TO 138
116     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 139
117     G=0.0
118     IF(I-IP)=142,142,143
         GO TO 143
119     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 144
120     G=0.0
121     IF(I-IP)=147,147,148
         GO TO 148
122     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 149
123     G=0.0
124     IF(I-IP)=152,152,153
         GO TO 153
125     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 154
126     G=0.0
127     IF(I-IP)=157,157,158
         GO TO 158
128     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 159
129     G=0.0
130     IF(I-IP)=162,162,163
         GO TO 163
131     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 164
132     G=0.0
133     IF(I-IP)=167,167,168
         GO TO 168
134     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 169
135     G=0.0
136     IF(I-IP)=172,172,173
         GO TO 173
137     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 174
138     G=0.0
139     IF(I-IP)=177,177,178
         GO TO 178
140     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 179
141     G=0.0
142     IF(I-IP)=182,182,183
         GO TO 183
143     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 184
144     G=0.0
145     IF(I-IP)=187,187,188
         GO TO 188
146     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 189
147     G=0.0
148     IF(I-IP)=192,192,193
         GO TO 193
149     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 194
150     G=0.0
151     IF(I-IP)=197,197,198
         GO TO 198
152     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 199
153     G=0.0
154     IF(I-IP)=202,202,203
         GO TO 203
155     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 204
156     G=0.0
157     IF(I-IP)=207,207,208
         GO TO 208
158     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 209
159     G=0.0
160     IF(I-IP)=212,212,213
         GO TO 213
161     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 214
162     G=0.0
163     IF(I-IP)=217,217,218
         GO TO 218
164     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 219
165     G=0.0
166     IF(I-IP)=222,222,223
         GO TO 223
167     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 224
168     G=0.0
169     IF(I-IP)=227,227,228
         GO TO 228
170     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 229
171     G=0.0
172     IF(I-IP)=232,232,233
         GO TO 233
173     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 234
174     G=0.0
175     IF(I-IP)=237,237,238
         GO TO 238
176     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 239
177     G=0.0
178     IF(I-IP)=242,242,243
         GO TO 243
179     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 244
180     G=0.0
181     IF(I-IP)=247,247,248
         GO TO 248
182     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 249
183     G=0.0
184     IF(I-IP)=252,252,253
         GO TO 253
185     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 254
186     G=0.0
187     IF(I-IP)=257,257,258
         GO TO 258
188     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 259
189     G=0.0
190     IF(I-IP)=262,262,263
         GO TO 263
191     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 264
192     G=0.0
193     IF(I-IP)=267,267,268
         GO TO 268
194     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 269
195     G=0.0
196     IF(I-IP)=272,272,273
         GO TO 273
197     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 274
198     G=0.0
199     IF(I-IP)=277,277,278
         GO TO 278
200     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 279
201     G=0.0
202     IF(I-IP)=282,282,283
         GO TO 283
203     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 284
204     G=0.0
205     IF(I-IP)=287,287,288
         GO TO 288
206     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 289
207     G=0.0
208     IF(I-IP)=292,292,293
         GO TO 293
209     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 294
210     G=0.0
211     IF(I-IP)=297,297,298
         GO TO 298
212     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 299
213     G=0.0
214     IF(I-IP)=302,302,303
         GO TO 303
215     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 304
216     G=0.0
217     IF(I-IP)=307,307,308
         GO TO 308
218     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 309
219     G=0.0
220     IF(I-IP)=312,312,313
         GO TO 313
221     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 314
222     G=0.0
223     IF(I-IP)=317,317,318
         GO TO 318
224     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 319
225     G=0.0
226     IF(I-IP)=322,322,323
         GO TO 323
227     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 324
228     G=0.0
229     IF(I-IP)=327,327,328
         GO TO 328
230     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 329
231     G=0.0
232     IF(I-IP)=332,332,333
         GO TO 333
233     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 334
234     G=0.0
235     IF(I-IP)=337,337,338
         GO TO 338
236     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 339
237     G=0.0
238     IF(I-IP)=342,342,343
         GO TO 343
239     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 344
240     G=0.0
241     IF(I-IP)=347,347,348
         GO TO 348
242     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 349
243     G=0.0
244     IF(I-IP)=352,352,353
         GO TO 353
245     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 354
246     G=0.0
247     IF(I-IP)=357,357,358
         GO TO 358
248     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 359
249     G=0.0
250     IF(I-IP)=362,362,363
         GO TO 363
251     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 364
252     G=0.0
253     IF(I-IP)=367,367,368
         GO TO 368
254     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 369
255     G=0.0
256     IF(I-IP)=372,372,373
         GO TO 373
257     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 374
258     G=0.0
259     IF(I-IP)=377,377,378
         GO TO 378
260     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 379
261     G=0.0
262     IF(I-IP)=382,382,383
         GO TO 383
263     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 384
264     G=0.0
265     IF(I-IP)=387,387,388
         GO TO 388
266     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 389
267     G=0.0
268     IF(I-IP)=392,392,393
         GO TO 393
269     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 394
270     G=0.0
271     IF(I-IP)=397,397,398
         GO TO 398
272     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 399
273     G=0.0
274     IF(I-IP)=402,402,403
         GO TO 403
275     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 404
276     G=0.0
277     IF(I-IP)=407,407,408
         GO TO 408
278     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 409
279     G=0.0
280     IF(I-IP)=412,412,413
         GO TO 413
281     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 414
282     G=0.0
283     IF(I-IP)=417,417,418
         GO TO 418
284     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 419
285     G=0.0
286     IF(I-IP)=422,422,423
         GO TO 423
287     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 424
288     G=0.0
289     IF(I-IP)=427,427,428
         GO TO 428
290     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 429
291     G=0.0
292     IF(I-IP)=432,432,433
         GO TO 433
293     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 434
294     G=0.0
295     IF(I-IP)=437,437,438
         GO TO 438
296     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 439
297     G=0.0
298     IF(I-IP)=442,442,443
         GO TO 443
299     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 444
300     G=0.0
301     IF(I-IP)=447,447,448
         GO TO 448
302     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 449
303     G=0.0
304     IF(I-IP)=452,452,453
         GO TO 453
305     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 454
306     G=0.0
307     IF(I-IP)=457,457,458
         GO TO 458
308     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 459
309     G=0.0
310     IF(I-IP)=462,462,463
         GO TO 463
311     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 464
312     G=0.0
313     IF(I-IP)=467,467,468
         GO TO 468
314     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 469
315     G=0.0
316     IF(I-IP)=472,472,473
         GO TO 473
317     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 474
318     G=0.0
319     IF(I-IP)=477,477,478
         GO TO 478
320     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 479
321     G=0.0
322     IF(I-IP)=482,482,483
         GO TO 483
323     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 484
324     G=0.0
325     IF(I-IP)=487,487,488
         GO TO 488
326     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 489
327     G=0.0
328     IF(I-IP)=492,492,493
         GO TO 493
329     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 494
330     G=0.0
331     IF(I-IP)=497,497,498
         GO TO 498
332     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 499
333     G=0.0
334     IF(I-IP)=502,502,503
         GO TO 503
335     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 504
336     G=0.0
337     IF(I-IP)=507,507,508
         GO TO 508
338     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 509
339     G=0.0
340     IF(I-IP)=512,512,513
         GO TO 513
341     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 514
342     G=0.0
343     IF(I-IP)=517,517,518
         GO TO 518
344     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 519
345     G=0.0
346     IF(I-IP)=522,522,523
         GO TO 523
347     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 524
348     G=0.0
349     IF(I-IP)=527,527,528
         GO TO 528
350     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 529
351     G=0.0
352     IF(I-IP)=532,532,533
         GO TO 533
353     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 534
354     G=0.0
355     IF(I-IP)=537,537,538
         GO TO 538
356     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 539
357     G=0.0
358     IF(I-IP)=542,542,543
         GO TO 543
359     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 544
360     G=0.0
361     IF(I-IP)=547,547,548
         GO TO 548
362     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 549
363     G=0.0
364     IF(I-IP)=552,552,553
         GO TO 553
365     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 554
366     G=0.0
367     IF(I-IP)=557,557,558
         GO TO 558
368     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 559
369     G=0.0
370     IF(I-IP)=562,562,563
         GO TO 563
371     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 564
372     G=0.0
373     IF(I-IP)=567,567,568
         GO TO 568
374     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 569
375     G=0.0
376     IF(I-IP)=572,572,573
         GO TO 573
377     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 574
378     G=0.0
379     IF(I-IP)=577,577,578
         GO TO 578
380     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 579
381     G=0.0
382     IF(I-IP)=582,582,583
         GO TO 583
383     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 584
384     G=0.0
385     IF(I-IP)=587,587,588
         GO TO 588
386     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 589
387     G=0.0
388     IF(I-IP)=592,592,593
         GO TO 593
389     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 594
390     G=0.0
391     IF(I-IP)=597,597,598
         GO TO 598
392     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 599
393     G=0.0
394     IF(I-IP)=602,602,603
         GO TO 603
395     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 604
396     G=0.0
397     IF(I-IP)=607,607,608
         GO TO 608
398     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 609
399     G=0.0
400     IF(I-IP)=612,612,613
         GO TO 613
401     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 614
402     G=0.0
403     IF(I-IP)=617,617,618
         GO TO 618
404     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 619
405     G=0.0
406     IF(I-IP)=622,622,623
         GO TO 623
407     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 624
408     G=0.0
409     IF(I-IP)=627,627,628
         GO TO 628
410     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 629
411     G=0.0
412     IF(I-IP)=632,632,633
         GO TO 633
413     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 634
414     G=0.0
415     IF(I-IP)=637,637,638
         GO TO 638
416     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 639
417     G=0.0
418     IF(I-IP)=642,642,643
         GO TO 643
419     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 644
420     G=0.0
421     IF(I-IP)=647,647,648
         GO TO 648
422     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 649
423     G=0.0
424     IF(I-IP)=652,652,653
         GO TO 653
425     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 654
426     G=0.0
427     IF(I-IP)=657,657,658
         GO TO 658
428     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 659
429     G=0.0
430     IF(I-IP)=662,662,663
         GO TO 663
431     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 664
432     G=0.0
433     IF(I-IP)=667,667,668
         GO TO 668
434     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 669
435     G=0.0
436     IF(I-IP)=672,672,673
         GO TO 673
437     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 674
438     G=0.0
439     IF(I-IP)=677,677,678
         GO TO 678
440     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 679
441     G=0.0
442     IF(I-IP)=682,682,683
         GO TO 683
443     G=PSI(1)*A(I+1,J)
         A(I+1,J)=A(I+1,J)-PSI(1)*E
         GO TO 684
444     G=0.0
445     IF(I-IP)=687,687,688
         GO TO 688
446     G=PSI(1)*A(I+
```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE D-T

73/74 OPT=1

FTN 4.2+75060 01/09/76 14.13.16.

```

      IF(I-IA)    34, 34, 30
54  A(I+?,J)=A(I+2,J)-PSI(?)*E
30  CONTINUE
      IF(I-IA)    35, 35, 35
31  L=I+?
32  GO TO 37
33  L=N
37  DO 40 J=10,L
      IF(I-IA)    38, 38, 39
38  C=PSI(?)*A(J,I+2)
      GO TO 41
39  C=0.0
40  E=A(1+?)*A(J,I)+PSI(1)*A(J,I+1)+C
      A(J,I)=A(J,I)-E
      A(J,I+1)=A(J,I+1)-PSI(1)*E
      IF(I-IA)    42, 42, 40
42  A(I+2)=A(I+2)-PSI(?)*E
43  CONTINUE
      IF(I-N+?)    43, 43, 100
43  =AL*PSI(2)*A(I+3,I+2)
      A(I+3,I)=-E
      A(I+3,I+1)=-PSI(1)*E
      A(I+3,I+2)=A(I+3,I+2)-PSI(?)*E
100  CONTINUE
101  RETURN
      END

```

QRT 9
QRT 10
QRT 11
QRT 12
QRT 13
QRT 14
QRT 15
QRT 16
QRT 17
QRT 18
QRT 19
QRT 20
QRT 21
QRT 22
QRT 23
QRT 24
QRT 25
QRT 26
QRT 27
QRT 28
QRT 29
QRT 30
QRT 31
QRT 32
QRT 33
QRT 34
QRT 35
QRT 36
QRT 37
QRT 38
QRT 39
QRT 40
QRT 41
QRT 42
QRT 43
QRT 44
QRT 45
QRT 46
QRT 47
QRT 48
QRT 49
QRT 50
QRT 51
QRT 52
QRT 53
QRT 54

SUBROUTINE RDISC

73774 CPY=1

FTN 4.2+75060

01/09/75 14:13:21.

```

      SUBROUTINE RDISC  (A,B,C,H,D,F,K1,K2,K3,K4,D,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      C
      C      THIS SUBROUTINE READS ALL INPUT MATRICES FROM DISCO ACCORDING
      C      TO THE PARAMETERS, SYSTEM AND OUTPUT, USING SUBROUTINE RDISCI
      C
      COMMON/COND/READ,SYSTEM,OUTPUT,NX,NY,NXC,NUC,N1,N2,DIGITL,
1CONTUR,NUMER,FPPS,TRESP,MONEL,NSCALE,SAV,CMAT,NK2,IFLAG,
?IGO,FORM,IPT,RFAD3,MIXED,MLTRT,SCAPLT,ZOH,KOUNT
      INTEGER READ,SYSTEM,OUTPUT,FORM,CONTUR,SAV,CMAT,READ3, FPPS,TRESP
      INTYCR   DIGITL,SCAPLT, ZCH
      DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),D(MY,MX),E(MY,MX),F(MY,MU),
1K1(MU,MX),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU)
      REAL K1,K2,K3,K4
      COMMON/SUBWRT/ ISURNAM
      IF(ITSUPNM.GE.?) WRITE(3,990)
      990 FORMAT (1X,*RDISC*)
      MAT1=MX
      MAT2=MY
      CALL RDISCI (A,NX,NX,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MU
      CALL RDISCI (B,NY,MU,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MY
      CALL RDISCI (C,NX,MX,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MU
      CALL RDISCI (K1,NU,NX,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL RDISCI (K2,NU,NX,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL RDISCI (K3,NU,NX,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL RDISCI (K4,NU,NX,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MY
      CALL RDISCI (H,NY,NX,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MU
      CALL RDISCI (I,NY,NU,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MU
      CALL RDISCI (D,NU,NU,
1MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      REWIND 8
      RETURN
      END
      RDISC    2
      RDISC    3
      RDISC    4
      RDISC    5
      RDISC    6
      RDISC    7
      RDISC    8
      RDISC    9
      RDISC   10
      RDISC   11
      RDISC   12
      RDISC   13
      RDISC   14
      RDISC   15
      RDISC   16
      RDISC   17
      RDISC   18
      RDISC   19
      RDISC   20
      RDISC   21
      RDISC   22
      RDISC   23
      RDISC   24
      RDISC   25
      RDISC   26
      RDISC   27
      RDISC   28
      RDISC   29
      RDISC   30
      RDISC   31
      RDISC   32
      RDISC   33
      RDISC   34
      RDISC   35
      RDISC   36
      RDISC   37
      RDISC   38
      RDISC   39
      RDISC   40
      RDISC   41
      RDISC   42
      RDISC   43
      RDISC   44
      RDISC   45
      RDISC   46
      RDISC   47
      RDISC   48
      RDISC   49
      RDISC   50
      RDISC   51

```

SUBROUTINE RDISCI 73/74 OPT=1

FTN 4.2+75060

01/09/71 14.13.53.

```
      SUBROUTINE RDISCI (A,N,M,
     1 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      DIMENSION A(MAT1,MAT2)
      COMMON/SUMWRT/ ISUBNAM
      IF(ISUBNAM.GE.2) WRITE(3,990)
 990  FORMAT(IX,*RDISCI*)
      DO 10 I=1,N
      READ (8) A(I,J),J=1,M
      IF(EOF(8).NE.0) STOP1
 10   CONTINUE
      RETURN
      END
```

RODISC1 ?
RODISC1 3
RODISC1 4
RODISC1 5
RODISC1 6
RODISC1 7
RODISC1 8
RODISC1 9
RODISC1 10
RODISC1 11
RODISC1 12
RODISC1 13

SUBROUTINE REDUCE 73/74 OPT#1 FTN 4.2+75760 01/03/78 14.13.55.

```

      SUBROUTINE REDUCE (IN,J,NM,4,A,B,C,ITEST,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      C
      C THIS SUBROUTINE DETERMINES THE IRREDUCIBLE SUBMATRICES OF
      C THE MATRIX A WITH DIMENSION N.
      C
      C J= NUMBER OF IRREDUCIBLE SUBMATRICES (1-5)
      C NM(I)= DIMENSION OF THE ITH SUBMATRIX
      C M= MATRIX WHOSE ITH ROW CONTAINS THE ORIGINAL ROW AND COLUMN
      C NUMBERS OF THE ITH SUBMATRIX
      C
      COMMON/COND/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,
     1CONTUR,NUMERS,FRPS,TRESP,MODEL,NSCALE,SAV,CMAT,NK2,IFLAG,
     2IGO,FORM,IPT,READ3,MIXED,MULTGT,SCAPLT,ZOM,KOUNT
      INTEGER READ,SYSTEM,OUTPUT,FORM,CONTUR,SAV,CMAT,READ3, FPRE,TRESP
      INTEGER DIGITL,SCAPLT,ZOM
      DIMENSION A(MX,MX),B(MX,MX),C(MX,MX)
      INTEGER SCAN
      DIMENSION M(10,20),MM(20),SCAN(20)
      COMMON/SUMWRT/ ISU3NAM
      IF (ISU3NAM.GE.2) WRITE(3,990)
 990 FORMAT(1X,*REDUCE*)
      DO 50 I=1,20
      DO 51 J=1,10
      M(I,J)=0
 51  CONTINUE
      MM(I)=0
 50  CONTINUE
      DO 10 I=1,N
      DO 11 J=1,N
      B(I,J)=0.0
      C(I,J)=0.0
      IF (A(I,J).NE.0.0) B(I,J)=1.0
 11  CONTINUE
      B(I,I)=1.0
 10  CONTINUE
      KRUNT=1
 22  CONTINUE
      DO 23 I=1,N
      DO 24 J=1,N
      DO 21 K=1,N
      IF (B(I,K).EQ.0.0.OR.B(K,J).EQ.0.0) GO TO 21
      C(I,J)=1.0
      GO TO 20
 21  CONTINUE
 20  CONTINUE
      KPRINT=2*KRUNT
      IF (KPRINT.GE.N-11 GO TO 23
      DO 24 I=1,N
      DO 24 J=1,N
      B(I,J)=C(I,J)
 24  CONTINUE
      GO TO 22
 23  CONTINUE
      DO 30 I=1,N
      SCAN(I)=I
 30  CONTINUE
      
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

      KEN
      DO 3 J=1,10
      MM(J)=0
      NK=1
      JJ=1
      DO 4 IF I=1,K
      IF (C(SCAN(1),SCAN(I)),EQ.1.0.AND.C(SCAN(I),SCAN(1)),EQ.1.0)
      1 GO TO 17
      GO TO 35
      37 M(J,JJ)=SCAN(I)
      MM(J)=MM(J)+1
      NK=NK+1
      38 CONTINUE
      K=K-NK
      IF (K,LT,0) GO TO 39
      LL=1
      39 KRINT=0
      KONT=0
      JX=K+NK
      DO 39 L=1,JX
      IF (SCAN(L),NE,M(J,LL)) GO TO 40
      KRINT=KRINT+1
      LL=LL+1
      GO TO 39
      40 KRINT=KRINT+1
      KCNT=KONT+1
      SCAN(KONT)=SCAN(KRINT)
      41 CONTINUE
      42 CONTINUE
      43 CONTINUE
      IF (K,GT,0) WRITE (3,100)
      100 FORMAT (//,* MORE THAN 10 REDUCIBLE SUBMATRICES*/)
      IF (K,GT,0) ITEST=1
      IF (IPT,LT,2) GO TO 200
      WRITE (3,201)
      201 FORMAT (/* MATRIX M IN REDUCE*/)
      C IN STATEMENT DO 35 J=1,10 IF THE LOOP GOES TO COMPLETION
      C J WILL = 11 ON CGC
      C J WILL = 10 ON IRM
      IF (J,GE,11) J=J-1
      DO 202 I=1,J
      MMM=MM(I)
      WRITE (3,203) (M(I,L),L=1,MMM)
      203 CONTINUE
      204 FORMAT (2015)
      200 CONTINUE
      RETURN
      END
      REDUCE   59
      REDUCE   60
      REDUCE   61
      REDUCE   62
      REDUCE   63
      REDUCE   64
      REDUCE   65
      REDUCE   66
      REDUCE   67
      REDUCE   68
      REDUCE   69
      REDUCE   70
      REDUCE   71
      REDUCE   72
      REDUCE   73
      REDUCE   74
      REDUCE   75
      REDUCE   76
      REDUCE   77
      REDUCE   78
      REDUCE   79
      REDUCE   80
      REDUCE   81
      REDUCE   82
      REDUCE   83
      REDUCE   84
      REDUCE   85
      REDUCE   86
      REDUCE   87
      REDUCE   88
      REDUCE   89
      REDUCE   90
      REDUCE   91
      REDUCE   92
      REDUCE   93
      REDUCE   94
      REDUCE   95
      REDUCE   96
      REDUCE   97
      REDUCE   98
      REDUCE   99
      REDUCE   100
      REDUCE 101
      REDUCE 102
      REDUCE 103
      REDUCE 104
      REDUCE 105
      REDUCE 106
      REDUCE 107

```

SUBROUTINE ROOT 73/74 OPT=1 FTN 4.2+7E06C 01/09/74 14.13.E4.
 1 U,V,
 2 >MX,MY,MU,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6
 3
 4 C THIS SUBROUTINE COMPUTES ROOT LOCII AS A FUNCTION OF TWO
 5 C FEEDBACK GAINS DEFINED BY K1,K2 AND K3,K4. THE OPEN LOOP
 6 C SYSTEM IS SCALED BY A SIMILARITY TRANSFORMATION. AT THE
 7 C CONCLUSION OF THE ROOT LOCII COMPUTATIONS THE NUMERATOR
 8 C ZEROS AND GAINS CORRESPONDING TO THE OPEN LOOP SYSTEM
 9 C ARE CALCULATED.
 10 C
 11 C N1= NUMBER OF ROOT LOCUS POINTS FOR THE K1,K2 GAIN
 12 C N2= NUMBER OF ROOT LOCUS POINTS FOR THE K3,K4 GAIN
 13 C
 14 C COMMON/COND/READ,SYSTEM,OUTPLT,NX,NY,NL,NXC,NUC,N1,N2,DIGITL,
 15 C ICONTUR,NUMERS,EPPS,TREF,TREFL,MODEL,NSCALE,SAV,CMAT,NK2,IFLAG,
 16 C ?IGC,FORM,IPT,READ3,MIXED,MULTRT,SCAPLT,ZOH,KCUNT
 17 C INTEGER READ,SYSTEM,OUTPUT,FORM,CONTUR,SAV,CMAT,READ3, EPPS,TREFH
 18 C INTGER DIGITL,SCAPLT,ZOH
 19 C REAL IFREQ,K1,K2,K3,K4,MN
 20 C COMMON/ACOND/DELT,FINAL,IFREQ,FREQ,OELFRO,GAIN1,GAIN2,MN
 21 C DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),D(MY,MX),E(MY,MX),F(MY,MU),
 22 C 1K1(MU),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU)
 23 C DIMENSION W1(MX,MX),W2(MX,MX),W3(MX,MX)
 24 C DIMENSION ROOTR(MX),ROOTI(MX),U(MX),V(MX)
 25 C DIMENSION M(10,20),MM(20),P(20)
 26 C COMMON/LABEL/INPT,OUTPT,TITLE
 27 C REAL INPT(10), OUTPT(20),TITLE(8)
 28 C COMMON/SUBWRIT/ ISUBNAM
 29 C DATA BLANK/AH /
 30 C DATA PST/10HPCLO /
 31 C IF (ISUBNAM.GE.2) WRITE(3,990)
 32 C IF (ISUBNAM .EQ. 2) WRITE (3,990)
 33 C 990 FORMAT(1X,*ROCT*)
 34 C IF (GAIN1.EQ.0.0) GAIN1=1.0
 35 C IF (GAIN2.EQ.0.0) GAIN2=1.0
 36 C WRITE (3,300) GAIN1,GAIN2
 37 C 300 FORMAT (/10X,*GAIN1 = *F10.4,10X,*GAIN2 = *F10.4/)
 38 C
 39 C 40 MAT1=MX
 40 MAT2=MX
 41 MAT3=MX
 42 MAT4=MX
 43 MAT5=MX
 44 MAT6=MX
 45
 46 IF (FORM.GT.0) WRITE(7)PST,TITLE,SYSTEM,MODEL,DIGITL,SCAPLT
 47 IF (FORM.GT.0) WRITE(7)N2,N1
 48 KK=1
 49 K1A=TAPE(N1)
 50 RR=0.0
 51 TT=0.0
 52 II=J
 53 JJ=0
 54 NK?=NK2+1
 55 DC 40 T=1,N2
 56 IF (N1.LT.0) GO TO 141
 57 II=I-1
 58 RR=(I-1)*GAIN2
 59

JMPUTTFE ROOT

73/74 OPT=1

FTN 4.2+7506

01/09/76 14.13.59.

```

      GO TO 142
141 RR=2.*PP
      II=2*IT
      IF (I.EQ.1) II=0
      IF (I.EQ.2) II=1
      IF (I.EQ.1) RR=0.0
      IF (I.EQ.2) RR=G4*TN2
142 CONTINUE
      DO 40 JJ=1,NIA
      IF (NI1.LT.0) GO TO 41
      JJ=J-1
      TT=(J-1)*GAIN1
      GO TO 42
41 TT=2.*TT
      JJ=2*JJ
      IF (J.EQ.1) JJ=0
      IF (J.EQ.2) JJ=1
      IF (J.EQ.1) TT=0.0
      IF (J.EQ.2) TT=GAIN1
42 CONTINUE
      GO TO (101,100),NK3
100 MAT1=MU
      MAT2=MX
      MAT3=MU
      MAT4=MX
      CALL ADD (TT,K2,RR,K4,W1,NU,NL,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MX
      MAT3=MX
      DO 43 IA=1,NU
      44 JA=1,NU
      W1(IA,JA)=W1(IA,JA)
44 CONTINUE
      W1(IA,JA)=1.0+W1(IA,IA)
43 CONTINUE
      CALL INVR (W1,W2,NU,R,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MU
      MAT3=MX
      CALL ADD (TT,K1,RR,K3,W1,NU,NX,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MX
      MAT3=MX
      CALL MULT (W2,W1,W3,NU,NU,NX,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      GO TO 102
101 MAT1=MU
      MAT2=MX
      MAT3=MU
      MAT4=MX
      CALL ADD (TT,K1,RR,K3,W3,NU,NX,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
102 CONTINUE
      IF (J.EQ.2.AND.I.EQ.1) GO TO 200
      IF (J.EQ.1.AND.I.EQ.2) GO TO 201
      GO TO 202
200 DO 203 IX=1,NU
      ROOT   59
      ROOT   60
      ROOT   61
      ROOT   62
      ROOT   63
      ROOT   64
      ROOT   65
      ROOT   66
      ROOT   67
      ROOT   68
      ROOT   69
      ROOT   70
      ROOT   71
      ROOT   72
      ROOT   73
      ROOT   74
      ROOT   75
      ROOT   76
      ROOT   77
      ROOT   78
      ROOT   79
      ROOT   80
      ROOT   81
      ROOT   82
      ROOT   83
      ROOT   84
      ROOT   85
      ROOT   86
      ROOT   87
      ROOT   88
      ROOT   89
      ROOT   90
      ROOT   91
      ROOT   92
      ROOT   93
      ROOT   94
      ROOT   95
      ROOT   96
      ROOT   97
      ROOT   98
      ROOT   99
      ROOT   100
      ROOT   101
      ROOT   102
      ROOT   103
      ROOT   104
      ROOT   105
      ROOT   106
      ROOT   107
      ROOT   108
      ROOT   109
      ROOT   110
      ROOT   111
      ROOT   112
      ROOT   113
      ROOT   114
      ROOT   115

```

SUBROUTINE ROOT 73/74 CPT=1

FTN 4.2+75260 01/09/76 14.13.59.

```

115      DC 205 JX=1,NX
          IF (NK2.EQ.1) GO TO 209
          H(KK,JX)=W3(IIX,JX)/TT
          GO TO 205
209  H(KK,JX)=W1(IIX,JX)/TT
204  CONTINUE
          IF (NK2.EQ.1) GO TO 203
          DO 207 JX=1,NU
          F(KK,JX)=K2(IIX,JX)
207  CONTINUE
125  KK=KK+1
          NY=NU
          GO TO 202
201  DC 204 IX=1,NU
          DC 206 JX=1,NX
130  IF (NK2.EQ.1) GO TO 210
          H(KK,JX)=W3(IIX,JX)/TT
          GO TO 206
210  H(KK,JX)=W1(IIX,JX)/RR
206  CONTINUE
          IF (NK2.EQ.0) GO TO 204
          DC 208 JX=1,NU
          F(KK,JX)=K4(IX,JX)
208  CONTINUE
204  KK=KK+1
          NY=2*NU
202  CONTINUE
          MAT1=MX
          MAT2=MU
          MAT3=MY
          MAT4=MY
          CALL MULT (R,W3,W2,NX,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
          MAT5=MX
          CALL ADD (1.0,A,1.0,M2,M1,NX,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
          IF (N2.NE.0) GO TO 220
          WRITE (3,221) J, TT
221  FORMAT (//,10X,*NO. *,I5,* ITERATION, 1ST GAIN =*,E12.4/
120X,* REAL PART*,15X,*IMAGINARY PART*)
          GO TO 222
220  WRITE (3,7) J, TT, I, RR
7  FORMAT (//,10X,*NO. *,I5,* ITERATION, 1ST GAIN =*,E12.4,10X,*NO. *
1,I5,* ITERATION, 2ND GAIN =*,E12.4/2DX,*REAL PART*,15X*
2*IMAGINARY PART*)
160  222 CONTINUE
          CALL EIGEN (NX,W1,M2,W3,ROOTR,ROCTI,U,V,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        45 CONTINUE
          DO 74 I=1,NU
          DO 74 J=1,NU
          D(I,J)=0.0
74  CONTINUE
          OUTPUT=3
          IF (NK2.EQ.0) OUTPUT=1
          FPSS=0
          NUMPRS=0

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE ROOT 73/74 OPT=1 FTN 4.2+75050 01/09/76 14.13.FN.

 175 OUTPT(1)=BLANK ROOT 173
 INPT(1)=BLANK ROOT 174
 NLU=NLU? ROOT 175
 DO 50 I=1,NLU ROOT 176
 DO 51 J=1,NX ROOT 177
 IF (H(I,J),EQ,0.0) GO TO 51 ROOT 178
 GO TO 52 ROOT 179
 51 CONTINUE ROOT 180
 50 CONTINUE ROOT 181
 52 IF (I,EQ,1) GO TO 54 ROOT 182
 IF (I,GT,NLU) I=NLU-NLU ROOT 183
 DO 53 J=1,NX ROOT 184
 H(I,J)=H(I,J)
 R(I,J)=R(J,I)
 53 CONTINUE ROOT 185
 DO 54 J=1,NU ROOT 186
 F(I,J)=F(I,J) ROOT 187
 56 CONTINUE ROOT 188
 54 NU=1 ROOT 189
 NU=1 ROOT 190
 WRITE (3,55) ROOT 191
 55 FORMAT (//,10X,*ROOT LCCUS OPTION COMPUTES THE ZEROES OF THE FIRST
 1 NON ZERO ROW#/10X,*OF K1 OR K3 RESPECTIVELY DUE TO INCIDATED ROOT 192
 ? INPUT */
 RETURN ROOT 193
 END ROOT 194
 195
 196
 197
 198

SUBROUTINE SETUP

73/74 OPT=1

FTN 4.2+75060

01/09/76 14.14.08.

```

SUBROUTINE SETUP (J,M,MM,P,A,B,C,H,G,F,K1,K2,K3,K4,D,W1,W2,W3,
1 R0CTR,
2 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)

C THIS SUBPROGRAM REDUCES THE ORIGINAL INPUT MATRICES FOR
C THE OPEN AND CLOSED LOOP SYSTEMS TO STANDARD FORM AND
C SCALES THE RESULTING SYSTEM WITH A DIAGONAL SIMILARITY
C TRANSFORMATION.
C DRAFTED BY G. NORRIS JULY 5 73
C
COMMON/COND/RPFD,SYSTEM,OUTPUT,NX,NY,NU,NKC,NUC,N1,N2,DIGITL,
1 CNTUR,NUMRS,FRPS,TRESP,MODEL,NSCALE,SAV,CHAT,NK2,TFLAG,
2 PIGO,FORM,IPT,READ3,MIXED,MULTRT,SCAPLT,ZOH,KOUNT
INTEGER READ,SYSTEM,OUTPUT,FORM,CNTUR,SAV,CHAT,READ3, FRPS, TRESP
COMMON/ACOND/ DELT,FINALT,IFREQ,FFREQ,DELFRQ,GAIN1,GAIN2,MMH
REAL K1, K2, K3, K4, IFREQ, MMH
DIMENSION H(10,20),MM(20),P(20)
DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),H(MY,MX),E(MY,MU),
1 K1(MU,MX),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU),
2 W1(MX,MX),W2(MX,MX),W3(MX,MX),R0CTR(MX)
DIMENSION W1(15,15), W2(15,15), W3(15,15)
C DIMENSION A(15,15), B(15,10), C(15,15), H(15,15), E(15,15)
C DIMENSION F(15,10), D(10,10), R0CTR(15),
1 K1(10,15), K2(10,15), K3(10,15), K4(10,15)
COMMON/RLKDAT/NUMBER,DENOM,GAIN,GRAPH,BLOCK,STATE,YTOV,ZTOU,YZTOK,
1 ITHINY,I THINU,NBLOCK,NZTCU,NYZTDK,NXT,NYT,NUT,NYI,NUI
REAL NUMBER
INTEGER GRAPH,BLOCK,STATE,YTOV,ZTCU,YZTOK
DIMENSION GRAPH(20,5),BLOCK(20,3),NUMBER(20,5),DENOM(20,5),
1 YGATN(PR),STATE(20,4),ITHINY(30),ITHINU(20),YTOV(20,2),
2 ZTOU(20,2),NYXU(4),YZTOK(20,2)
COMMON/SLBWRIT/ ISUBNM
IF(IUBRNAM.GE.2) WRITE(3,990)
990 FORMAT(1X,*SETUP*)
IM1MX=1
ICHE=0
MAT1=MY
MAT2=MX
MAT3=MX
MAT4=MX
MAT5=MX
MAT6=MX
IF (CMAT.EQ.0) GO TO 5
C
C ELIMINATE C MATRIX
C
CALL INVR (C,W2,NX,1,
1 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MULT (W2,A,W1,NX,NX,NX,
1 MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MAKE (A,W1,NX,NX,
1 MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT4=MU
CALL MULT (W2,B,W1,NX,NX,NU,
1 MX,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT?=MU

```

```

      MAT4=MX
      CALL MAKE (R,W1,NX,NU,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      DMATE=0
      % GC TC (F0,10,10), SYSTEM
      10 IF (NK2.EQ.0) GO TO 40

      15
      16      ELIMINATE K2,K4 MATRICES
      17
      18      IF (SYSTEM.EQ.3.AND.MIXED.NE.1) GO TO 660
      19      MAT1=MU
      20      MAT2=MX
      21      MAT4=MU
      22      CALL MULT (K2,R,W1,NU,NX,NU,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      23      DO 12 I=1,NU
      24      DO 13 J=1,NU
      25      W2(I,J)=W1(I,J)
      26      13 CONTINUE
      27      W2(I,I)=1.0+W2(I,I)
      28      12 CONTINUE
      29      IF (DCITL.NE.1) GO TO 838
      30      DO 31 I=1,NU
      31      DO 32 J=1,NU
      32      W2(I,J)=0.0
      33      IF (I.EQ.J) W2(I,I)=1.0
      34      39 CONTINUE
      35      838 CALL INVR (W2,W1,NU,0,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      36      MAT1=MU
      37      MAT4=MX
      38      CALL MULT (K2,A,W2,NU,NX,NX,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      39      CALL ADD ((1.0,K1,1.0,W2,W3,NU,NX,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      40      MAT1=MX
      41      CALL MULT (W1,W3,W2,NU,NU,NY,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      42      MAT3=MU
      43      MAT4=MU
      44      CALL MULT (W1,D,W3,NU,NU,NU,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      45      MAT1=MU
      46      MAT2=MU
      47      MAT3=MX
      48      MAT4=MX
      49      CALL MAKE (D,W3,NU,NU,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      50      MAT2=MX
      51      CALL MAKE (K1,W2,NU,NX,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      52      NK2=0
      53      % GC TO 40
      560 MAT1=MU
      570 MAT2=MX
      580 CALL MULT (K2,A,W1,NU,NX,NX,
      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)

```

```

117      CALL MULT (K4,A,W2,NU,NX,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL ADD (1.0,K1,1.0,W1,W3,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL ADD (1.0,K3,1.0,W2,W1,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MAKE (K1,W3,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MAKE (K3,W1,MU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MULT (K4,P,W1,NU,NX,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MULT (K4,B,W2,NU,NX,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT4=MX
CALL MAKE (K2,W1,NU,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MAKE (K4,W2,NU,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
135    +) CONTINUE
50 IF (COUTPUT.EQ.1.OR.OUTPUT.F0.3) GO TO 51
C
C   ELIMINATE G MATRIX
C
140      MAT1=MY
MAT2=MX
MAT3=MY
MAT4=MU
CALL MULT (G,P,W1,NY,NX,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT2=MU
MAT4=MX
MAT3=MX
CALL ADD (1.0,F,1.0,W1,W2,NY,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MAKE (F,W2,NY,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT2=MX
CALL MULT (G,A,W1,NY,NX,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL ADD (1.0,H,1.0,W1,W2,NY,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MAKE (H,W2,NY,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
IF (COUTPUT.EQ.2) OUTPUT=3
IF (COUTPUT.EQ.4) OUTPUT=3
51 IF (RMIXED.NE.1.OR.PREAD.EQ.-1) GO TO 210
C
C   AUGMENT SYSTEM WITH CONTROL SYSTEM MODELED IN CLASS
C
165      MAT1=MY
MAT2=MX
CALL ZOT1 (C,MY,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
DO 211 I=1,NY
C(I,I)=1.0

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE: S-TUP 77/74 OPT=1

FTN 4.2+75060 01/09/74 14.14.08.

```

    711 CONTINUE
    CALL CLASS (A,R,C,H,G,F,D,W1,W2,W3,
    1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
    712 CONTINUE
    MAT1=MX
    MAT2=MY
    MAT3=MX
    MAT4=MX
    CALL MAKF (W1,A,NX,NY,
    1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
    IF (INSCALE.EQ.0) GO TO 52
    IF (INSCALE.EQ.2) GO TO 52
    C
    C   SYSTEM SCALED BY A DIAGONAL SIMILARITY TRANSF-MAT1 N
    C
    ICK=L
    CALL ASCALE (NX,J,M,MM,P,W1,W2,W3,ROCR,
    1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
    DO 50 I= 1,NX
    DO 50 K= 1,NU
    K1(K,I)=K1(K,I)/F(I)
    IF (SYSTEM.NE.3.AND.NK2.NE.1) GO TO 661
    K3(K,I)=K3(K,I)/F(I)
    R1(I,K)=R1(I,K)*P(I)
    50 CONTINUE
    IF (SYSTEM.EQ.3.OR.MIXE.Q.1) GO TO 662
    DO 70 I=1,NY
    DO 70 K= 1,NX
    H(I,K)=H(I,K)/P(K)
    70 CONTINUE
    662 MAT1=MX
    MAT2=MX
    CALL MAKF (A,W1,NX,NY,
    1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
    IF (IPT.LT.1) GO TO 52
    WRITE (3,110)
    110 FORMAT (1H1,10X,*TIME REDUCED AND SCALED SYSTEM IS*)
    CALL SFIT (A,P,C,H,G,F,K1,K2,K3,K4,D,
    1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
    52 IF (DGTTL.NE.1) GO TO 80
    IF (NTTOL.EQ.0.AND.NYTOV.EQ.0) GO TO 465
    IMIX=1
    GO TO 40
    215 C
    C   DISCRETIZE CCNTINUOUS PORTION OF SAMPLED-DATA SYSTEM
    C
    465 IF (MMM.EQ.0.0) GO TO 455
    IF (SYSTEM.NE.1) GO TO 456
    GO TO 457
    220 456 WRITE (3,458)
    458 FORMAT (1/* ROOT LOCUS AND CLOSED LOOP OPTIONS NOT ALLOWED WITH NO
    -MODIFIED Z-TRANSFORMS*/)
    457 XM=MMN*DELT
    CALL EAT (YM,A,W1,W2,W3,C,NYC,
    1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
    IF (ZOH.EQ.0) GO TO 459
    OUTPUT=3
    SETUP 173
    SETUP 174
    SETUP 175
    SETUP 176
    SETUP 177
    SETUP 178
    SETUP 179
    SETUP 180
    SETUP 181
    SETUP 182
    SETUP 183
    SETUP 184
    SETUP 185
    SETUP 186
    SETUP 187
    SETUP 188
    SETUP 189
    SETUP 190
    SETUP 191
    SETUP 192
    SETUP 193
    SETUP 194
    SETUP 195
    SETUP 196
    SETUP 197
    SETUP 198
    SETUP 199
    SETUP 200
    SETUP 201
    SETUP 202
    SETUP 203
    SETUP 204
    SETUP 205
    SETUP 206
    SETUP 207
    SETUP 208
    SETUP 209
    SETUP 210
    SETUP 211
    SETUP 212
    SETUP 213
    SETUP 214
    SETUP 215
    SETUP 216
    SETUP 217
    SETUP 218
    SETUP 219
    SETUP 220
    SETUP 221
    SETUP 222
    SETUP 223
    SETUP 224
    SETUP 225
    SETUP 226
    SETUP 227
    SETUP 228
    SETUP 229

```

```

240      MAT1=MY
        CALL MULT (H,H2,W3,YY,NYC,NYC,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        MAT1=MU
        MAT1=MY
        CALL MULT (W3,P,C,YY,NYC,NUC,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        MAT1=MY
        MAT1=MY
        MAT1=MU
        MAT1=MY
        MAT1=MU
        CALL ADD (1.,C,1.,F,F,YY,NUC,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        MAT1=MY
        MAT1=MY
        MAT1=MY
        MAT1=MY
        MAT1=MY
        CALL MULT (H,W1,W3,YY,NYC,NYC,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        CALL MAKE (H,W3,YY,NYC,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        IF (ZOH.EQ.NUC) GO TO 455
        NN1=ZOH+1
        DO 450 I=1,NY
        DO 450 J=NN1,NLC
        IF (F(I,J).NE.0) GO TO 461
450 CONTINUE
        GO TO 462
461 WRITE (3,448)
462 DO 463 I=1,NY
        DO 463 J=NN1,NUC
        XX=0.0
        DO 464 K=1,NYC
        XX=XX+H(I,K)*B(K,J)
464 CONTINUE
        F(I,J)=XX
463 CONTINUE
        OUTPUT=3
465 CONTINUE
        CALL EAT (DELT,A,W1,W2,W3,C,NYC,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        NN1=ZOH+1
        DO 7 I=1,NYC
        DO 7 J=1,NYC
        A(I,J)= W1(I,J)
7 CONTINUE
        IF (ZOH.EQ.0) GO TO 441
        MAT1=MY
        MAT1=MU
        CALL MULT (W2,B,W3,NYC,NYC,ZOH,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        MAT2=MU
        MAT2=MY
        CALL MAKE (B,W3,NYC,ZOH,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)

```

SETUP 230
SETUP 231
SETUP 232
SETUP 233
SETUP 234
SETUP 235
SETUP 236
SETUP 237
SETUP 238
SETUP 239
SETUP 240
SETUP 241
SETUP 242
SETUP 243
SETUP 244
SETUP 245
SETUP 246
SETUP 247
SETUP 248
SETUP 249
SETUP 250
SETUP 251
SETUP 252
SETUP 253
SETUP 254
SETUP 255
SETUP 256
SETUP 257
SETUP 258
SETUP 259
SETUP 260
SETUP 261
SETUP 262
SETUP 263
SETUP 264
SETUP 265
SETUP 266
SETUP 267
SETUP 268
SETUP 269
SETUP 270
SETUP 271
SETUP 272
SETUP 273
SETUP 274
SETUP 275
SETUP 276
SETUP 277
SETUP 278
SETUP 279
SETUP 280
SETUP 281
SETUP 282
SETUP 283
SETUP 284
SETUP 285
SETUP 286

SUBROUTINE SH-TUP 73/74 CPT=1 FTN 4.2+75060 01/09/76 14.14.08.

 441 IF (ZOH.FD.NUC) GO TO 442
 442 IF (ZOH.LT.NUC) GO TO 443
 443 WRITF (3,444)
 444 FORMAT (1,* ZOH GREATER THAN NUC*)
 290 445 DO 445 I=1,NY
 446 DO 445 J=NN1,NUC
 447 IF (F(I,J).NE.0.0) GO TO 448
 448 CONTINUE
 449 GO TO 447
 295 450 WRITE (3,448)
 448 FORMAT (1,* F MATRIX NOT NULL *)
 449 DO 449 I=1,NYC
 450 DO 449 J=NN1,NUC
 451 XX=0.0
 300 DO 450 K=1,NXC
 452 XX=XX+H(I,K)*R(K,J)
 453 CONTINUE
 454 H3(I,J)= XX
 449 CONTINUE
 305 IF (MMM.NE.0.0) GO TO 470
 455 DO 455 I=1,NY
 456 DO 455 J=NN1,NUC
 457 XX=0.0
 310 DO 458 K=1,NXC
 459 XX=XX+H(I,K)* R(K,J)
 460 CONTINUE
 461 F(I,J) = XX
 462 CONTINUE
 470 DO 470 I=1,NYC
 471 DO 471 J=NN1,NUC
 472 R(I,J)= H3(I,J)
 473 CONTINUE
 474 OUTPUT=3
 475 CONTINUE
 320 476 IF (IPT.LT.1) GO TO 80
 477 WRITE (3,111)
 111 FORMAT (1H1,10X,* THE DESCRETIZED SYSTEM IS*)
 CALL SPIT (A, B, C, H, G, F, K1, K2, K3, K4, D,
 1MX, MY, MU, MS, MAT1, MAT2, MAT3, MAT4, MAT5, MAT6)
 325 478 CONTINUE
 479 IF (IMX.EQ.0) GO TO 500
 480 IF (IMX.EQ.2) GO TO 500
 C C SURSYSTEMS DESCRIBED IN STEP1 AND STEP2 OF MIXED LOADING OPTION
 340 C ARE COUPLED USING YTOV AND ZTCU
 C
 481 IF (INBLOCK.EQ.0) GO TO 500
 482 MAT1=MX
 483 MAT2=MY
 484 MAT3=MX
 485 MAT4=MY
 486 CALL ZCT1 (M1, NUT, NYT,
 1MX, MY, MU, MS, MAT1, MAT2, MAT3, MAT4, MAT5, MAT6)
 487 IF (INZTOU.EQ.0) GO TO 500
 488 DO 489 I=1,NZTCU
 489 W1(ZTOU(I,2),ZTOU(I,1)+NY1)=1.0
 490 CONTINUE

SETUP 287
 SETUP 288
 SETUP 289
 SETUP 290
 SETUP 291
 SETUP 292
 SETUP 293
 SETUP 294
 SETUP 295
 SETUP 296
 SETUP 297
 SETUP 298
 SETUP 299
 SETUP 300
 SETUP 301
 SETUP 302
 SETUP 303
 SETUP 304
 SETUP 305
 SETUP 306
 SETUP 307
 SETUP 308
 SETUP 309
 SETUP 310
 SETUP 311
 SETUP 312
 SETUP 313
 SETUP 314
 SETUP 315
 SETUP 316
 SETUP 317
 SETUP 318
 SETUP 319
 SETUP 320
 SETUP 321
 SETUP 322
 SETUP 323
 SETUP 324
 SETUP 325
 SETUP 326
 SETUP 327
 SETUP 328
 SETUP 329
 SETUP 330
 SETUP 331
 SETUP 332
 SETUP 333
 SETUP 334
 SETUP 335
 SETUP 336
 SETUP 337
 SETUP 338
 SETUP 339
 SETUP 340
 SETUP 341
 SETUP 342
 SETUP 343

	SUBROUTINE	73/74	OPT=1	FTN 4, 2+7506P	01/09/74	14.14.05.
	300 IF (NYTOV.EQ.0) GO TO 302			SETUP	344	
	DO 303 I=1,NYTCV			SETUP	345	
345	W1(YTOV(I,2)+NU1,YTOV(I,1))=1.0			SETUP	346	
	303 CONTINUE			SETUP	347	
	MAT1=MY			SETUP	348	
	MAT2=MU			SETUP	349	
350	302 CALL MULT (F,W1,W3,NYT,NUT,NYT,			SETUP	350	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	351	
	DO 304 I=1,NYT			SETUP	352	
	DO 307 J=1,NYT			SETUP	353	
	W3(I,J)=W3(I,J)			S-TUP	354	
355	307 CONTINUE			SETUP	355	
	W3(I,I)=1.0+W3(I,I)			SETUP	356	
356	306 CONTINUE			SETUP	357	
	CALL INV(R,W3,C,NYT,1,			SETUP	358	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	359	
360	MAT1=MX			SETUP	360	
	MAT2=MX			SETUP	361	
	MAT3=MY			SETUP	362	
	CALL MULT (C,H,W3,NYT,NYT,NXT,			SETUP	363	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	364	
	MAT3=MY			SETUP	365	
	MAT4=MY			SETUP	366	
	CALL MAKE (H,W3,NYT,NXT,			S-TUP	367	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	368	
	IF (ICK,NE.1) GO TO 312			SETUP	369	
370	DO 313 I=1,NY			SETUP	370	
	DO 313 J=1,NX			SETUP	371	
	H(I,J)=H(I,J)/P(J)			SETUP	372	
	313 CONTINUE			SETUP	373	
371	312 CONTINUE			S-TUP	374	
	MAT1=MX			SETUP	375	
	MAT2=MY			SETUP	376	
	MAT3=MY			SETUP	377	
	MAT4=MU			SETUP	378	
380	CALL MULT (C,F,W3,NYT,NUT,			SETUP	379	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	380	
	MAT1=MY			SETUP	381	
	MAT2=MU			SETUP	382	
	MAT3=MX			SETUP	383	
	MAT4=MX			SETUP	384	
385	CALL MAKE (F,W3,NYT,NUT,			SETUP	385	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	386	
	MAT1=MY			SETUP	387	
	MAT2=MU			SETUP	388	
	CALL MULT (B,W1,W3,NXT,NUT,NYT,			SETUP	389	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	390	
390	MAT2=MY			SETUP	391	
	MAT3=MY			SETUP	392	
	CALL MULT (W3,H,W1,NXT,NYT,NXT,			SETUP	393	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	394	
	MAT3=MY			SETUP	395	
	CALL ADD (1.0,A,1.0,W1,A,NXT,NYT,			SETUP	396	
	1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)			SETUP	397	
	MAT3=MY			SETUP	398	
	MAT4=MU			SETUP	399	
395	CALL MULT (W3,F,W1,NXT,NYT,NUT,			SETUP	400	

ORIGINAL PAGE IS
OF POOR QUALITY

DEBOUTING N TUP 73/74 OPT=1

FTN 4.2+7506

01/09/76 14.14.08.

```

470      1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6
        IF (ICK,NE,1) GO TO 310
        DO 311 I=1,NX
        DO 311 J=1,MY
        W1(I,J)=W1(I,J)/P(J)
310      T11 CONTINUE
310      T10 CONTINUE
        MAT1=MX
        MAT2=MY
        MAT3=MY
        MAT4=MY
        MAT5=MY
        MAT6=MY
        CALL ADD (1.0,P,-1.0,W1,W2,NXT,NUT,
        1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        MAT7=MY
        CALL MAE (0,W2,NXT,NUT,
        1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
500      CONTINUE
        IF (ITMIN,NF, 1) GO TO 468
        IMTX=?
        GO TO 465
465      IF (IPRAN,EQ,4,OR,MIXED,EQ,1) GO TO 728
        GO TO 721
728      IF (NYZTOK,EO,0) GO TO 720
C
C      CONSTRUCT FEEDBACK MATRICES K1 AND K3 FROM YZTCK
425      C
        IF (SYSTM,EO,1) SYSTEM=?
        K=1
        I=1
710      MAT1=MX
        MAT2=MY
        CALL ZCT1(W1,NUT,NX,
        1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
        CALL ZCT1(W2,NUT,NUT,
        1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
435      T13 CONTINUE
        DO 714 J=1,NX
        W1(YZTCK(I,1),J)= W1(YZTOK(I,1),J) +W1(YZTOK(I,2),J)
714      CONTINUE
        DO 713 J=1,NUT
        W2(YZTOK(I,2),J)= F(YZTOK(I,1),J) +W2(YZTOK(I,2),J)
715      CONTINUE
        I=I+1
        IF (SYSTM ,EO, 2) GO TO 730
        DO 900 J=1,NYZTOK
        DO 901 L=1,20
        IF (ITHINU(21-L),EO,0) GO TO 802
        IF (YZTOK(J,2),GT,ITHINU(21-L)) GO TO 803
        ITHINU(22-L)=ITHINU(21-L)
901      CONTINUE
904      FORMAT (*10X,*N0. *,15,*, INPUT MUST BE SAVED FOR FEEDBACK*)
        DO 802 L=1,20
        IF (ITHINU(21-L),EO,0) GO TO 802
        IF (YZTOK(J,2),GT,ITHINU(21-L)) GO TO 803
        ITHINU(22-L)=ITHINU(21-L)
802      CONTINUE
803      ITHINU(22-L)=YZTOK(J,2)
900      CONTINUE

```

SUBROUTINE D-TRP 73/74 CPT=1 FTN 4.2+75060 01/09/76 14.14.0H.

 450
 MAT1=MU
 MAT2=MU
 IF (K .LE. 2) GO TO 731
 CALL MAKF (K1,M1,NUT,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 CALL MAKF (K2,M2,NUT,NUT,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 IF (NY2TOK .EQ. 1) GO TO 732
 K=2
 GO TO 716
 731 CALL MAKF (K3,W1,NUT,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 CALL MAKF (K4,W2,NUT,NUT,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 IF (NY2TOK .GT. 2) WRITE (3,722)
 722 FORMAT (/*ONLY 2 LOOPS ALLOWED*
 732 DO 733 L=1,NUT
 DO 733 J=1,NUT
 IF (K2(L,J) .NE. 0.0) GO TO 734
 IF (K4(L,J) .NE. 0.0) GO TO 734
 733 CONTINUE
 GO TO 720
 734 NK=1
 GO TO 720
 730 IF (I .LE. NY2TOK) GO TO 713
 DO 735 J=1,NUT
 DO 737 L=1,NUT
 H2(I,L)=W2(J,L)
 737 CONTINUE
 H2(J,I)=1.0+H2(I,J)
 736 CONTINUE
 MAT1=MX
 MAT2=MX
 451
 738 CALL TNV0 (W2,W3,NUT,1,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 MAT1=MU
 MAT2=MU
 MAT3=MY
 MAT4=MX
 CALL MAKF (W1,W2,NUT,NUT,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 MAT1=MY
 MAT2=MU
 CALL MULT (W1,W2,NUT,NUT,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 MAT1=MU
 CALL MAKF (K1,W2,NUT,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 500
 720 CONTINUE
 IF (DIGITL .NE. 1) GO TO 6
 DO A37 I=1,NU
 DO A37 J=1,NUC
 K2(I,J)=0.0
 K4(I,J)=0.0
 837 CONTINUE
 6 CONTINUE
 IF (MULTPT .NE. 0) GO TO 727

	SETUP	458
	SETUP	459
	SETUP	460
	SETUP	461
	SETUP	462
	SETUP	463
	SETUP	464
	SETUP	465
	SETUP	466
	SETUP	467
	SETUP	468
	SETUP	469
	SETUP	470
	SETUP	471
	SETUP	472
	SETUP	473
	SETUP	474
	SETUP	475
	SETUP	476
	SETUP	477
	SETUP	478
	SETUP	479
	SETUP	480
	SETUP	481
	SETUP	482
	SETUP	483
	SETUP	484
	SETUP	485
	SETUP	486
	S-TUP	487
	SETUP	488
	SETUP	489
	SETUP	490
	S-TUP	491
	SETUP	492
	SETUP	493
	SETUP	494
	SETUP	495
	SETUP	496
	SETUP	497
	SETUP	498
	SETUP	499
	SETUP	500
	SETUP	501
	SETUP	502
	SETUP	503
	SETUP	504
	SETUP	505
	SETUP	506
	SETUP	507
	SETUP	508
	SETUP	509
	SETUP	510
	SETUP	511
	SETUP	512
	SETUP	513
	SETUP	514

SURVEYING SECTION

7317-0 QP:1

FTN 4,2+75060

01/09/76 14.14.08.

```

IF (SYSTEM .NE. 2) GO TO 727          SETUP      515
FOR CLOSFD LOOP ANALYSIS, INCOPORATE FEEDBACK MATRX,K1
AND FEEDFORWARD MATRX,C               SETUP      516
SETUP      517
SETUP      518
SETUP      519
SETUP      520
SETUP      521
SETUP      522
SETUP      523
SETUP      524
SETUP      525
SETUP      526
SETUP      527
SETUP      528
SETUP      529
SETUP      530
SETUP      531
SETUP      532
SETUP      533
SETUP      534
SETUP      535
SETUP      536
SETUP      537
SETUP      538
SETUP      539
SETUP      540
SETUP      541
SETUP      542
SETUP      543
SETUP      544
SETUP      545
SETUP      546
SETUP      547
SETUP      548
SETUP      549
SETUP      550
SETUP      551
SETUP      552
SETUP      553
SETUP      554
SETUP      555
SETUP      556
SETUP      557
SETUP      558
SETUP      559
SETUP      560
SETUP      561
SETUP      562
SETUP      563
SETUP      564
SETUP      565
SETUP      566
SETUP      567
SETUP      568
SETUP      569
SETUP      570
SETUP      571

MAT1=MX
MAT2=MU
MAT3=MU
MAT4=NY
CALL MULT (B,K1,W1,NX,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT2=NX
MAT3=MX
CALL ADD (1.0,A,1.0,M1,W2,NX,NY,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MAKE (A,W2,NX,NX,
1MX,MU,MY,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT2=MU
MAT3=MU
MAT4=MU
CALL MULT (B,D,W1,NX,NU,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT2=MX
MAT3=NY
CALL MAKE (B,W1,NX,NU,
1MX,MU,MY,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT1=MY
MAT2=NU
MAT3=MU
CALL MULT (F,K1,W1,NY,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT2=NX
MAT3=MX
CALL ADD (1.0,H,1.0,W1,W2,NY,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
CALL MAKE (H,W2,NY,NX,
1MX,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT2=NU
MAT3=MU
MAT4=NU
CALL MULT (F,D,W1,NY,NU,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
MAT3=NX
MAT4=MX
CALL MAKE (F,W1,NY,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
SYSTEM=1
727 CONTINUE
IF (PTAD.EQ.4.AND.SYSTEM.EQ.3) GO TO 828
IF (MIXEDINE.1) GO TO 721
828 IF (NBLOCK.EQ.0) GO TO 520

THIN THE INPUT VECTOR (MIXED LOADING OPTION)
DO 394 I=1,10
DO 321 J=1,NX
R(J,I)=R(J,I*THINU(I))
394
321

```

SUBROUTINE C7TUP 73/74 OPT=1 F7N 4.2+75060 01/69/76 14.14.08.
 121 CONTINUE
 DO 323 J=1,NY
 F(J,I)=F(J,I*THINU(I))
 323 CONTINUE
 IF (KYSTEM .EQ. 3 .OR. MULTOT .NE. 0) GO TO 322
 GO TO 305
 322 DO 325 J=1,NX
 K1(I,J)=K1(I*THINU(I),J)
 K3(I,J)=K3(I*THINU(I),J)
 325 CONTINUE
 DO 326 J=1,NUT
 D(J,I)=D(J,I*THINU(I))
 326 CONTINUE
 DO 327 J=1,NUT
 E(I,J)=E(I*THINU(I),J)
 327 CONTINUE
 305 IF (I*THINU(I+1) .EQ. 0) GO TO 320
 304 CONTINUE
 320 IF (I.LT.NUT) NY=I
 721 CONTINUE
 TF (.WAD,EQ.4.0R.MIXED.EQ.1.0R.READ,EQ.3) GO TO 729
 GO TO 520
 725 CONTINUE
 C THIN THE OUTPUT VECTOR (MIXED LOADING OPTION)
 C
 DO 501 I=1,30
 DO 502 J=1,NXT
 H(I,J)=H(I*THINY(I),J)
 512 CONTINUE
 DO 503 J=1,NUT
 F(I,J)=F(I*THINY(I),J)
 505 CONTINUE
 IF (I*THINY(I+1) .EQ. 0) GO TO 521
 501 CONTINUE
 21 IF (I.LT.NY) NY=I
 IF (IMTX.NE.1) GO TO 520
 IMTX=?
 GO TO 465
 520 MAT1=MX
 MAT2=MX
 MAT3=MX
 MAT4=MX
 CALL MAKE (W1,A,NX,NX,
 1NY,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 IF (MIXED.EQ.1) GO TO 102
 IF (IPY.LT.1) GO TO 200
 102 WRITE (*,100)
 100 FORMAT(1H1,10X,* THE FINAL REDUCED SYSTEM IS//)
 CALL SPIT (A,R,C,H,G,F,K1,K2,K3,K4,D,
 1NY,NY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 200 CONTINUE
 ENDO

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE SPIT

13/74 CPT=1

FTN 4, 2475160

01/09/76 14.15.00.

```

SUBROUTINE SPIT  (A,N,C,H,F,K1,K2,K3,K4,D,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      THIS SUBROUTINE PRINTS ALL MATRICES ACCORDING TO THE
      PARAMETERS, SYSTEM AND OUTPLT, LSING SUBROUTINE SPIT1
      COMMON/DOND/READ,SYSTEM,OUTPLT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,
1CCNTUR,NUM,RS,FRPS,TRESP,MODL,NSCALE,SAV,CMAT,NK2,IFLAG,
2IGO,FORM,IPT,READY,MIXED,MULTRT,SCAPLT,ZDM,KCUNT
      INTEGER READ,SYSTEM,OUTPUT,FORM,CONTUR,SAV,CMAT,READY,FRPS,TRESP
      INTEGER DIGITL,SCAPLT,ZCH
      DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),H(MY,MX),G(MY,MX),F(MY,MU),
1K1(MU,MX),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU)
      REAL K1,K2,K3,K4
      COMMON/SUBRIT/ TSJUNAM
      IF(TSJUNAM.GE.2) WRITE(3,990)
990 FORMAT(1X,*SPIT*)
      MAT1=MX
      MAT2=MX
      MAT3=MX
      MAT4=MX
      MAT5=MX
      MAT6=MX
      WRITE(3,10)
10 FORMAT(// 10X,*THE A MATRIX IS*)
      CALL SPIT1 (A,NX,NK,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      WRITE(3,11)
11 FORMAT(// 10X,*THE B MATRIX IS*)
      MAT2=MU
      CALL SPIT1 (B,NX,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MX
      IF (CMAT.EQ.0) GO TO 21
      WRITE(3,12)
12 FORMAT(// 10X,*THE C MATRIX IS*)
      CALL SPIT1 (C,NX,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      21 CONTINUE
      GC TO (50,50,60),SYSTEM
      50 WRITE(3,13)
13 FORMAT(// 10X,*THE K1 MATRIX IS*)
      MAT1=MU
      CALL SPIT1 (K1,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF (NK2.EQ.0) GO TO 62
      WRITE(3,13)
13 FORMAT(// 10X,*THE K2 MATRIX IS*)
      CALL SPIT1 (K2,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      62 CONTINUE
      IF (NK2.EQ.0) GO TO 64
      WRITE(3,14)
14 FORMAT(// 10X,*THE K3 MATRIX IS*)
      CALL SPIT1 (K3,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      IF (NK2.EQ.0) GO TO 64
      WRITE(3,15)
15 FORMAT(// 10X,*THE K4 MATRIX IS*)
      CALL SPIT1 (K4,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      SPIT      2
      SPIT      3
      SPIT      4
      SPIT      5
      SPIT      6
      SPIT      7
      SPIT      8
      SPIT      9
      SPIT     10
      SPIT     11
      SPIT     12
      SPIT     13
      SPIT     14
      SPIT     15
      SPIT     16
      SPIT     17
      SPIT     18
      SPIT     19
      SPIT     20
      SPIT     21
      SPIT     22
      SPIT     23
      SPIT     24
      SPIT     25
      SPIT     26
      SPIT     27
      SPIT     28
      SPIT     29
      SPIT     30
      SPIT     31
      SPIT     32
      SPIT     33
      SPIT     34
      SPIT     35
      SPIT     36
      SPIT     37
      SPIT     38
      SPIT     39
      SPIT     40
      SPIT     41
      SPIT     42
      SPIT     43
      SPIT     44
      SPIT     45
      SPIT     46
      SPIT     47
      SPIT     48
      SPIT     49
      SPIT     50
      SPIT     51
      SPIT     52
      SPIT     53
      SPIT     54
      SPIT     55
      SPIT     56
      SPIT     57
      SPIT     58

```

SUBROUTINE SPIT 73774 CPT=1 FTN 4,2+75060 01/09/75 14.15.00
 60 CONTINUE
 IF (MX=0,FO,1) GO TO 50
 GO TO 200
 50 WRITE (3,16)
 16 F0=MAT (// 10X,*THE H MATRIX IS*)
 17 F0=MAT (// 10X,*THE G MATRIX IS*)
 18 F0=MAT (// 10X,*THE F MATRIX IS*)
 19 F0=MAT (// 10X,*THE D MATRIX IS*)
 MAT1=MY
 CALL SPIT1 (H,NY,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 GO TO (100,56,57,58),OUTPUT
 56 WRITE (3,17)
 CALL SPIT1 (G,NY,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 GO TO 100
 57 WRITE (3,18)
 MAT2=MU
 CALL SPIT1 (F,NY,NU,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 GO TO 100
 58 WRITE (3,17)
 CALL SPIT1 (G,NY,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 WRITE (3,18)
 MAT2=MU
 CALL SPIT1 (F,NY,NU,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 100 GO TO (200,110,200),SYSTEM
 110 WRITE (3,12)
 MAT1=MU
 MAT2=MY
 CALL SPIT1 (K1,NU,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 IF (NK2,FO,0) GO TO 66
 WRITE (3,13)
 CALL SPIT1 (K2,NU,NX,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 66 CONTINUE
 WRITE (3,19)
 MAT2=MU
 CALL SPIT1 (D,NU,NU,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 200 CONTINUE
 RETURN
 END

SUBROUTINE SPIT1 73/74 CPT=1

FTN 4.2+75060 01/09/76 14:15:49.

```
SUBROUTINE SPIT1 (A,N,M,
1 MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
DIMENSION A(MAT1,MAT2)
COMMON/SUBMRT1/ ISURNAM
IF(ITSUINAH,GE,2) WRITE(3,940)
990 FORMAT(1X,*SPIT1*)
WRITE(3,10) N,M
10 FORMAT(2I10/)
DO 20 I=1,N
20 CONTINUE
30 FORMAT(10E12.4)
RETURN
END
```

	SPIT1	?
	SPIT1	3
	SPIT1	4
	SPIT1	5
	SPIT1	6
	SPIT1	7
	SPIT1	8
	SPIT1	9
	SPIT1	10
	SPIT1	11
	SPIT1	12
	SPIT1	13
	SPIT1	14
	SPIT1	15

```

      SUBROUTINE THIST (A,R,C,H,F,W1,W2,W3,ROOTR,ROOTI,U,K1,D,Z,V,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)

      THIS SUBROUTINE COMPUTES A TABULATED TIME HISTORY RESPONSE
      OF THE SYSTEM. INPUT TO THE SYSTEM IS UNDER USER CONTROL
      THROUGH A CALL TO SUBROUTINE INPUT (DELT,T,U) WHERE DELT IS
      THE TIME INCREMENT, T IS THE TIME, AND U IS THE INPUT VECTOR
      OF DIMENSION MX. THE RESPONSE IS COMPUTED USING THE STATE
      TRANSITION MATRIX. THE TRANSITION MATRIX IS COMPUTED BY THE
      SUBROUTINE FAT USING THE SERIES EXPANSION TECHNIQUE. TEN TERMS
      OF THE SERIES ARE USED.

      CORRECTION MADE BY G. NORRIS JULY 5 73

      CCMCN/COND/RFAO,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,CIGITL,
     1CCNTUR,NUMERS,FRES,TRESP,MODEL,NSCALE,SAV,CHAT,NK2,IFLAG,
     2IGO,FORM,IPR,READ3,MIXFC,MULTRT,SCAPLT,ZOH,KCLNT
      CCMCN/ACOND/ DELT,FINALT,IFREQ,FFREQ,DELFRQ,GAIN1,GAIN2,M
      INTEGER READ,SYSTEM,OUTPUT,FORM,CCNTUR,SAV,CHAT,READ3, FPRS,TRESP
      INTEGER DIGITL,SCAPLT,ZOH
      REAL K1, IFREQ, M
      COMMON/LABEL/INPT,OUTPT,TITLE
      REAL INPT(10), OUTPT(20),TITLE(2)
      DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),H(MY,MX),F(MY,MU),
     1W1(MX,MX),W2(MX,MX),W3(MX,MX),ROOTR(MX),ROOTI(MX),U(MX),
     2K1(MU,MX),D(MU,MU),Z(MX),V(MX)
      DIMENSION A(15,15), B(15,15), C(15,15), H(15,15)
      DIMENSION F(15,10),K1(10,15), D(10,10), Z(15), V(15), U(15)
      DIMENSION W1(15,15), W2(15,15), W3(15,15)
      DIMENSION ROOTR(15), ROOTI(15)
      DIMENSION K2(10,15), K3(10,15), K4(10,15)
      REAL PST
      COMMON/SUBWRIT/ ISUBNAM
      DATA PST/10HTIME /
      IF(IISUNAM,GE,2) WRITE(3,990)
      990 FORMAT(1X,THIST*)
      IF (DIGITL,NE,0) GO TO 60
      IF (MULTRT,GT,0) GO TO 60
      CALL FAT (DELT,A,W1,W2,W3,C,NX,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MX
      MAT2=MX
      MAT3=MX
      MAT4=MX
      MAT5=MX
      MAT6=MX
      CALL MAKF (A,W1,NX,MX,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT6=MU
      CALL MULT (W2,B,W3,NX,NY,NU,
     1MY,YY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MU
      MAT3=MX
      CALL MAKF (B,W1,NX,NU,
     1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      60 CONTINUE
      DC 400 TR=1,TRESP
      T=0.0

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE THICKT 73/74 OP1=1 FTN 4,2+75060 01/09/70 14.15.51.

 60 DO 1 I=1,NX
 ROOTP(I)=0.0
 ROOT(I)=0.0
 V(I)=0.0
 W1(I)=0.0
 U(I)=0.0
 65 1 CONTINUE
 NXU=NNU+1
 NUU=NNU+1
 IF(FORM,GT,0) WRITE(7) PST,TITLE,SYSTEM,MODEL,DIGITL,SCAPLT
 NYU=NY+NU
 70 IF (MULTRT,EQ,1) NYU=NY+NUC
 IF(FORM,GT,0) WRITE(7) NYU,(CUTFT(I),I=1,NY),(INPT(J),J=1,NU)
 WRITE(7,13) (OUTP(T(I),I=1,NY),(INP(T(J),J=1,NU)
 13 FORMAT (//, TIME *(13(2X,A10))//)
 IF (MULTRT,NE,0) GO TO 98
 75 20 CONTINUE
 CALL INPUTV(DELT,T,U,
 1MX,MY,NU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 DO 3 J=1,NY
 RCTT(I)=0.0
 80 DO 30 J=1,NX
 ROOT(I)=ROOT(I)+H(I,J)*RCTT(J)
 30 CONTINUE
 DO 31 I=1,NY
 DO 31 J=1,NU
 RCTT(I)=ROOT(I)*W(T,J)*U(J)
 31 CONTINUE
 IF(FORM,EQ,2) GO TO 6
 WRITE(7,10) T,(ROOT(I),I=1,NY),(U(J),J=1,NU)
 IF(FORM,EQ,0) GO TO 7
 90 5 WRITE(7) T,(ROOT(I),I=1,NY),(U(J),J=1,NU)
 7 CONTINUE
 10 FORMAT (1E12.3)
 DO 40 I=1,NX
 W1(I,1)=0.0
 95 DO 40 J=1,NX
 W1(I,J)=W1(I,1)+A(I,J)*RCTR(J)
 40 CONTINUE
 DO 41 I=1,NX
 DO 41 J=1,NU
 W1(I,J)=W1(I,1)+B(I,J)*U(J)
 41 CONTINUE
 DO 42 I=1,NX
 RCTR(I)=W1(I,1)
 42 CONTINUE
 135 T=T+DT
 IF (T,LE,FINALT) GO TO 20
 GO TO 96
 98 KCT=1
 ICK=0
 110 120 CONTINUE
 IF (ICK,FO,1) GO TO 70
 CALL INPUTV(DELT,T,U,
 1MX,MY,NU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 DO 7F I=1,NXC

60	SPIT1	73
	SPIT1	74
	SPIT1	75
	SPIT1	76
	SPIT1	77
	SPIT1	78
	SPIT1	79
	SPIT1	80
	SPIT1	81
	SPIT1	82
	SPIT1	83
	SPIT1	84
	SPIT1	85
	SPIT1	86
	SPIT1	87
	SPIT1	88
	SPIT1	89
	SPIT1	90
	SPIT1	91
	SPIT1	92
	SPIT1	93
	SPIT1	94
	SPIT1	95
	SPIT1	96
	SPIT1	97
	SPIT1	98
	SPIT1	99
	SPIT1	100
	SPIT1	101
	SPIT1	102
	SPIT1	103
	SPIT1	104
	SPIT1	105
	SPIT1	106
	SPIT1	107
	SPIT1	108
	SPIT1	109
	SPIT1	110
	SPIT1	111
	SPIT1	112
	SPIT1	113
	SPIT1	114
	SPIT1	115
	SPIT1	116
	SPIT1	117
	SPIT1	118
	SPIT1	119
	SPIT1	120
	SPIT1	121
	SPIT1	122
	SPIT1	123
	SPIT1	124
	SPIT1	125
	SPIT1	126
	SPIT1	127
	SPIT1	128
	SPIT1	129

SUBROUTINE	ROUTINE	73/74	OPT=1	FTN 4.2+7C96	01/09/76	14.16.51.
115	V(I)=RCDFR(I)			SPIT1	130	
	76 CONTINUE			SPIT1	131	
	DO 71 I=1,NU			SPIT1	132	
	XX=0.			SPIT1	133	
	DO 72 J=1,NU			SPIT1	134	
	XX=XX+K1(I,J)*U(J)			SPIT1	135	
120	77 CONTINUE			SPIT1	136	
	Z(I)=XX			SPIT1	137	
	78 CONTINUE			SPIT1	138	
	DO 73 I=1,NU			SPIT1	139	
	XX=0.			SPIT1	140	
	DO 74 J=1,NX			SPIT1	141	
	XX=XX+K1(I,J)*RCOTR(J)			SPIT1	142	
125	74 CONTINUE			SPIT1	143	
	Z(I)=Z(I)+XX			SPIT1	144	
130	73 CONTINUE			SPIT1	145	
	ICX=1			SPIT1	146	
	70 CONTINUE			SPIT1	147	
	DO 80 I=1,NY			SPIT1	148	
	XX=0.			SPIT1	149	
135	DO 81 J=1,NYC			SPIT1	150	
	XX=XX+K1(I,J)*RCOTR(J)			SPIT1	151	
	81 CONTINUE			SPIT1	152	
	RCOTI(I)=XX			SPIT1	153	
140	80 CONTINUE			SPIT1	154	
	DO 82 I=1,NY			SPIT1	155	
	XX=0.			SPIT1	156	
	DO 83 J=1,NYC			SPIT1	157	
	XX=XX+K1(I,J)*Z(J)			SPIT1	158	
145	83 CONTINUE			SPIT1	159	
	RCOTI(I)=RCOTI(I)+XX			SPIT1	160	
	82 CONTINUE			SPIT1	161	
	IF (F0M4.EQ.2) GO TO 16			SPIT1	162	
	WRITE (3,10) T,(ROOTI(I),I=1,NY),(Z(J),J=1,NYC)			SPIT1	163	
150	16 WRITE (7) T, (ROOTI(I),I=1,NY), (Z(J),J=1,NYC)			SPIT1	164	
	17 CONTINUE			SPIT1	165	
	DO 85 I=1,NYC			SPIT1	166	
	XX=0.			SPIT1	167	
	DO 86 J=1,NYC			SPIT1	168	
	XX=XX+ALI(J)*RCOTR(J)			SPIT1	169	
155	86 CONTINUE			SPIT1	170	
	W1(I,1)=XX			SPIT1	171	
	87 CONTINUE			SPIT1	172	
	DO 87 I=1,NYC			SPIT1	173	
160	XX=0.			SPIT1	174	
	DO 88 J=1,NYC			SPIT1	175	
	XX=XX+R(T,J)*Z(J)			SPIT1	176	
	88 CONTINUE			SPIT1	177	
	W1(I,1)=W1(I,1)+XX			SPIT1	178	
165	87 CONTINUE			SPIT1	179	
	DO 89 I=1,NYC			SPIT1	180	
	ROOTR(I)=W1(I,1)			SPIT1	181	
	89 CONTINUE			SPIT1	182	
	T=T+DELT			SPIT1	183	
	KCT=KCT+1			SPIT1	184	
170	IF (T.GT.FINALT) GO TO 96			SPIT1	185	
				SPIT1	186	

ROUTINE	LINE	73/74 OPT=1	FTN 4.2+75060	01/09/76 14:15:51.
		IF (KCT, LE, MULTRT) GO TO 120	SPIT1	157
		KCT=1	SPIT1	158
		ICK=0	SPIT1	159
120		IF (NXX, GT, NX) GO TO 120	SPIT1	160
		DO 30 I=NXX, NX	SPIT1	161
		XX=0.	SPIT1	162
		DO 91 J=NXX, NX	SPIT1	163
		XX=XX+A(I,J)*PCOTR(J)	SPIT1	164
120	91	CONTINUE	SPIT1	165
		W1(I,I)=XX	SPIT1	166
		90 CONTINUE	SPIT1	167
		DO 92 I=NXX, NX	SPIT1	168
		XX=1.	SPIT1	169
120	92	DO 93 J=1,NXC	SPIT1	170
		XX=XX+A(I,J)*V(J)	SPIT1	171
		93 CONTINUE	SPIT1	172
		W1(I,I)=W1(I,I)+XX	SPIT1	173
		92 CONTINUE	SPIT1	174
140		IF (NUU, GT, NU) GO TO 130	SPIT1	175
		DO 94 I=NXX, NX	SPIT1	176
		XX=0.	SPIT1	177
		DO 95 J=NUU, NU	SPIT1	178
		XX=XX+R(I,J)*Z(J)	SPIT1	179
140	95	CONTINUE	SPIT1	180
		RCTR(I)=W1(I,I)+XX	SPIT1	181
		96 CONTINUE	SPIT1	182
		GO TO 120	SPIT1	183
230	130	DO 131 I=NXX, NX	SPIT1	184
		ROOTR(I)=W1(I,I)	SPIT1	185
	131	CONTINUE	SPIT1	186
		GO TO 120	SPIT1	187
	96	TT=-1.0	SPIT1	188
		IF (FORM, GT, 0) WRITE(7) TT,	SPIT1	189
205		XTT, TT, TT	SPIT1	220
		96 CONTINUE	SPIT1	221
		END	SPIT1	222

SUBROUTINE SH7

73/74 OPT=1

FTN 4.2475050

01/09/76 14.15.59.

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE SW7 73/74 OPT=1

FTN 4,2+75160 01/09/76 14.15.59.

	GO TO 100	SWZ	59
103	IF (NIP(I,7).EQ.1) GO TO 120	SWZ	60
	IF (NIP(I,7).EQ.2) GO TO 121	SWZ	61
	BLOCK(L,2)=1	SWZ	62
	BLOCK(L,3)=2	SWZ	63
	NUMER(L,1)=1.	SWZ	64
	DENOM(L,1)=0.	SWZ	65
	DENOM(L,2)=1.	SWZ	66
	GO TO 100	SWZ	67
	120 GAIN(L)=GAIN(L)*.5*DELT	SWZ	68
	121 BLOCK(L,2)=2	SWZ	69
	BLOCK(L,3)=2	SWZ	70
70	NUMER(L,1)=1.	SWZ	71
	NUMER(L,2)=1.	SWZ	72
	DENOM(L,1)=1.	SWZ	73
	DENOM(L,2)=1.	SWZ	74
	GO TO 100	SWZ	75
75	104 IF (NIP(I,7).EQ.1) GO TO 141	SWZ	76
	IF (NIP(I,7).EQ.2) GO TO 142	SWZ	77
	BLOCK(L,2)=1	SWZ	78
	BLOCK(L,3)=2	SWZ	79
	NUMER(L,1)=1.	SWZ	80
	DENOM(L,1)=1.	SWZ	81
	DENOM(L,2)=1./PARAM(I,2)	SWZ	82
	GO TO 100	SWZ	83
	141 PARAM(I,2)=TANH(.5*PARAM(I,2)*DELT)	SWZ	84
	142 XNU=PARAM(I,2)	SWZ	85
85	BLOCK(L,2)=2	SWZ	86
	BLOCK(L,3)=2	SWZ	87
	NUMER(L,1)=XNU/(1.+XNU)	SWZ	88
	NUMER(L,2)=NUMER(L,1)	SWZ	89
	DENOM(L,2)=1.	SWZ	90
	DENOM(L,1)=(XNU-1.)/(XNU+1.)	SWZ	91
	GO TO 100	SWZ	92
	105 BLOCK(L,2)=2	SWZ	93
	BLOCK(L,3)=2	SWZ	94
	IF (NIP(I,7).EQ.1) GO TO 151	SWZ	95
	IF (NIP(I,7).EQ.2) GO TO 152	SWZ	96
	NUMER(L,1)=1.	SWZ	97
	NUMER(L,2)=1./PARAM(I,3)	SWZ	98
	DENOM(L,1)=1.	SWZ	99
	DENOM(L,2)=1./PARAM(I,2)	SWZ	100
100	GO TO 100	SWZ	101
	151 PARAM(I,2)=TANH(.5*PARAM(I,2)*DELT)	SWZ	102
	PARAM(I,3)=TANH(.5*PARAM(I,3)*DELT)	SWZ	103
	152 XNU=PARAM(I,2)	SWZ	104
	XNU=PARAM(I,3)	SWZ	105
	DENOM(L,1)=(XNU-1.)/(XNU+1.)	SWZ	106
	DENOM(L,2)=1.	SWZ	107
	NUMER(L,1)=(XNU*(XNU-1.))/(XNU*(XNU+1.))	SWZ	108
	NUMER(L,2)=(XNU*(XNU+1.))/(XNU*(XNU+1.))	SWZ	109
	GO TO 100	SWZ	110
105	176 BLOCK(L,2)=2	SWZ	111
	BLOCK(L,3)=2	SWZ	112
	IF (NIP(I,7).EQ.1) GO TO 161	SWZ	113
	IF (NIP(I,7).EQ.2) GO TO 162	SWZ	114
	NUMER(L,1)=0.	SWZ	115

```

115      NUMFR(L,2)=1.
      DENOM(L,1)=PARAM(I,2)
      DENOM(L,2)=1.
      GO TO 100
120      161 PARAM(I,2)=TANH(.5*PARAM(I,2)*DELT)
      162 XN=PARAM(I,2)
      NUMFR(L,1)=-1./(1.+XN)
      NUMFR(L,2)=-NUMER(L,1)
      DENOM(L,1)=(XN-1.)/(XN+1.)
      DENOM(L,2)=1.
      GO TO 100
125      107 IF (NIP(I,7).EQ.1) GO TO 171
      IF (NIP(I,7).EQ.2) GO TO 172
      BLOCK(L,2)=1
      BLOCK(L,3)=3
130      171 NUMFR(L,1)=1.
      DENOM(L,3)=1./(PARAM(I,2)*PARAM(I,3))
      DENOM(L,2)=1./PARAM(I,2)+1./PARAM(I,3)
      DENOM(L,1)=1.
      GO TO 100
135      172 PARAM(I,3)=TANH(.5*PARAM(I,2)*DELT)
      PARAM(I,2)=TANH(.5*PARAM(I,3)*DELT)
      173 XNU=PARAM(I,2)
      XN=PARAM(I,3)
      GAIN(L)=GAIN(L)*XN*XNU/((XN+1.)*(XNU+1.))
      BLOCK(L,2)=3
      BLOCK(L,3)=3
      NUMFR(L,1)=1.
      NUMFR(L,2)=2.
      NUMFR(L,3)=1.
140      141 DENOM(L,1)=(XN-1.)*(XNU-1.)/(XN+1.)*(XNU+1.)
      DENOM(L,2)=(XN-1.)/(XN+1.)+(XNU-1.)/(XNU+1.)
      DENOM(L,3)=1.
      GO TO 100
145      108 IF (NIP(I,7).EQ.1) GO TO 181
      IF (NIP(I,7).EQ.2) GO TO 182
      BLOCK(L,2)=1
      BLOCK(L,3)=3
      NUMFR(L,1)=1.
      DENOM(L,3)=1./(PARAM(I,2)**2)
      DENOM(L,2)=2.*PARAM(I,3)/PARAM(I,2)
      DENOM(L,1)=1.
      GO TO 109
150      181 ALFA=PARAM(I,2)*PARAM(I,3)
      RTA=SQR(PARAM(I,2)**2-ALFA**2)
      ALFA=ALFA*DELT
      RTA=RTA*LT
      D=(EXP(A)*COS(B)+1.)*2+EXP(B)**2*EXP(2.*A)
      U=(EXP(2.*A)-1.)/D
      V=E**2*SIN(A)*EXP(A)/D
      W=SQR(U**2+V**2)
      OMEGA=PARAM(I,2)
      ZETA=PARAM(I,3)
      PARAM(I,2)=W
      PARAM(I,3)=-U/W
155      182 BLOCK(L,2)=3
      BLOCK(L,3)=3
      SWZ   116
      SWZ   117
      SWZ   118
      SWZ   119
      SWZ   120
      SWZ   121
      SWZ   122
      SWZ   123
      SWZ   124
      SWZ   125
      SWZ   126
      SWZ   127
      SWZ   128
      SWZ   129
      SWZ   130
      SWZ   131
      SWZ   132
      SWZ   133
      SWZ   134
      SWZ   135
      SWZ   136
      SWZ   137
      SWZ   138
      SWZ   139
      SWZ   140
      SWZ   141
      SWZ   142
      SWZ   143
      SWZ   144
      SWZ   145
      SWZ   146
      SWZ   147
      SWZ   148
      SWZ   149
      SWZ   150
      SWZ   151
      SWZ   152
      SWZ   153
      SWZ   154
      SWZ   155
      SWZ   156
      SWZ   157
      SWZ   158
      SWZ   159
      SWZ   160
      SWZ   161
      SWZ   162
      SWZ   163
      SWZ   164
      SWZ   165
      SWZ   166
      SWZ   167
      SWZ   168
      SWZ   169
      SWZ   170
      SWZ   171
      SWZ   172

```

SUBROUTINE SWZ 73/74 OPT=1 F7N 4.2+75060 01/09/76 14.15.59.

 X=PARAM(I,2)
 DENOM(L,1)=(1.-2.*PARAM(I,3)*XN+XN**2)/(1.+2.*PARAM(I,3)*XN+XN**2)
 DENOM(L,2)=2.*(XN**2-1.)/(1.+2.*PARAM(I,3)*XN+XN**2)
 DENOM(L,3)=1.
 GAIN(L)=GAIN(L)*XN**2/(1.+2.*PARAM(I,3)*XN+XN**2)
 109 IF (NIP(I,7).NE.0) GO TO 110
 II=NIP(I,2)-7
 GO TO (100,221,222,223),II
 140 221 BLOCK(L,2)=3
 NUMER(L,3)=1./PARAM(I,4)**2
 NUMER(L,2)=2.*PARAM(I,5)/PARAM(I,4)
 NUMER(L,1)=1.
 GO TO 100
 185 222 BLOCK(L,2)=2
 NUMER(L,2)=1./PARAM(I,4)
 NUMER(L,1)=1.
 GO TO 100
 223 BLOCK(L,2)=2
 NUMER(L,1)=0.
 NUMER(L,2)=1.
 GO TO 100
 110 II=NIP(I,2)-7
 GO TO (230,232,231,230),II
 195 231 IF (NIP(I,7).EQ.2) GO TO 230
 PARAM(I,4)=TANH(.5*PARAM(I,4)*DELT)
 GO TO 230
 232 IF (NIP(I,7).EQ.2) GO TO 230
 ALFA=-PARAM(I,4)*PARAM(I,5)
 BETA=SORT(PARAM(I,4)**2-ALFA**2)
 A=ALFA*DELT
 B=BETA*DELT
 D=(EXP(A)*COS(B)+1.)**2+SIN(B)**2*EXP(2.*A)
 U=(EXP(2.*A)-1.)/D
 205 V=2.*SIN(B)*EXP(A)/D
 W=SQRT(U**2+V**2)
 PARAM(I,4)=W
 PARAM(I,5)=-U/W
 230 GO TO (240,241,242,243),II
 210 240 NUMER(L,1)=1.
 NUMER(L,2)=2.
 NUMER(L,3)=1.
 GO TO 100
 241 YN=PARAM(I,4)
 NUMER(L,2)=2.*(YN**2-1.)/(1.+2.*PARAM(I,5)*YN+YN**2)
 NUMER(L,1)=(1.-2.*PARAM(I,5)*YN+YN**2)/(1.+2.*PARAM(I,5)*YN+YN**2)
 NUMER(L,3)=1.
 GAIN(L)=GAIN(L)*(1.+2.*PARAM(I,5)*YN+YN**2)/YN**2
 GO TO 100
 220 242 YN=PARAM(I,4)
 NUMER(L,1)=(YN-1.)/(YN+1.)
 NUMER(L,2)=2.*YN/(YN+1.)
 NUMER(L,3)=1.
 GAIN(L)=GAIN(L)*(1.+YN)/YN
 GO TO 100
 225 243 NUMER(L,1)=-1.
 NUMER(L,2)=0.
 NUMER(L,3)=1.

SUBROUTINE G4Z 73/74 CPT=1 FTN 4,2+75060 01/09/76 14.1E.59.

```
240      IF (NIP(I,7).EQ.2) GO TO 100
      GAIN(L)=GAIN(L)*(1.+2.*PARAM(I,3)*XN+XN**2)/XN**2
      GAIN(L)=GAIN(L)*(1.+DENOM(L,1)+DENOM(L,2))
      1/(2.*DFLT)
100  CONTINUE
      RETURN
      END
```

SWZ 230
 SWZ 231
 SWZ 232
 SWZ 233
 SWZ 234
 SWZ 235
 SWZ 236

ORIGINAL PAGE IS
OF POOR QUALITY

FUNCTION TANG 73/74 CPT=1

FTN 4.2+75060 01/09/78 14.19.0E.

```
      FUNCTION TANG (A)
      COMPL-X A,C
      EQUIVALENCE (C,B(1))
      DIMENSION B(2)
      5   C=A
      PHI=ATAN2 (B(2),B(1))*57.3
      IF (B(2).GT.0.0.AND.B(1).LT.0.0) GO TO 10
      TANG=PHI
      GO TO 20
10   TANG=PHI-180.
20   RETURN
      END
```

TANG	2
TANG	3
TANG	4
TANG	5
TANG	6
TANG	7
TANG	8
TANG	9
TANG	10
TANG	11
TANG	12
TANG	13

SUBROUTINE WDISC 73/74 OPT=1 FTN 4.2+75060 01/09/76 14:19:07.

```

      SUBROUTINE WDISC (A,B,C,H,G,F,K1,K2,K3,K4,D,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      C
      C THIS SUBROUTINE WRITES ALL INPUT MATRICES ON DISCS ACCORDING
      C TO THE PARAMETERS, SYSTEM AND OUTPUT, USING SUBROUTINE WDISC1
      C
      COMMON/COND/READ,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,
1CONTUP,NUMR7,FRFS,TRESP,MOOSL,NSCAL7,SAV,CHAT,NK2,IFLAG,
2IGO,FORM,IPT,READ3,MIXED,MULTRT,SCAPLT,ZOH,KOUT
      INTEGER READ,SYSTEM,OUTPUT,FORM,CONTUP,SAV,CHAT,READ3, FPPS,TRESP
      INTEGER DIGITL,SCAPLT,70H
      DIMENSION A(MX,MX),B(MX,MU),C(MX,MU),D(MY,MX),E(MY,MY),F(MY,MU),
1K1(MU,MX),K2(MU,MX),K3(MU,MX),K4(MU,MX),D(MU,MU)
      REAL K1,K2,K3,K4
      COMMON/SUBWRIT/ ISUB3NAM
      IF(ISUB3NAM.GE.2) WRITE(3,9901
990 FORM(1X,*WDISC*)
      MAT1=MY
      MAT2=MX
      CALL WDISC1 (A,NX,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MU
      CALL WDISC1 (B,NX,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MX
      CALL WDISC1 (C,NX,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MU
      CALL WDISC1 (K1,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL WDISC1 (K2,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL WDISC1 (K3,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL WDISC1 (K4,NU,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MY
      CALL WDISC1 (H,NY,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL WDISC1 (G,NY,NX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MY
      CALL WDISC1 (F,NY,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=NU
      CALL WDISC1 (D,NU,NU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      REWIND 8
      RETURN
      END

```

SUBROUTINE WDISC1 73/74 CPT=1

FTN .. 2+75360 01/09/76 14.19.11.

```
      SUBROUTINE WDISC1 (A,N,M,
 1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 2      DIMENSION A(MAT1,MAT2)
 3      CCMMCN/SUMMPIT/ TSURKAM
 4      IF(ISURNAM.GE.2) WRITE(3,990)
 5      990 FORMAT(1X,*WDISC1*)
 6      DO 10 I=1,N
 7      WRITE(8) (A(I,J),J=1,M)
 8      10 CONTINUE
 9      RETURN
10      END
```

	WDISC1	2
5	WDISC1	3
	WDISC1	4
	WDISC1	5
	WDISC1	6
	WDISC1	7
	WDISC1	8
10	WDISC1	9
	WDISC1	10
	WDISC1	11
	WDISC1	12

SUBROUTINE ZOT 73/74 CPT=1 FTN 4.2+7E06C 01/09/76 14.19.13.

```

      SUBROUTINE ZOT (A,B,C,H,G,F,K1,K2,K3,K4,D,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      COMMON/COND/PFAO,SYSTEM,OUTPUT,NX,NY,NU,NXC,NUC,N1,N2,DIGITL,
1CNTUR,NUMRS,FRPS,TRFSP,MODEL,NSCALE,SAV,CMAT,NK2,IFLAG,
?LOC,FORM,IPT,READ3,MIXED,MULTRT,SCAPLT,ZOH,KCUNT
      INTEGER (IGITL,SCAPLT,ZOH
      INTEGER READ,SYSTEM,OUTPUT,FORM,CNTUR,SAV,CMAT,READ3, FRPS,TRFSP
      ZOT   2
      ZOT   3
      ZOT   4
      ZOT   5
      ZOT   6
      ZOT   7
      ZOT   8
      ZOT   9
      REAL K1,K2,K3,K4
      DIMENSION A(MX,MX),B(MX,MU),C(MX,MX),D(MY,MX),E(MY,MU),
1K1(MU,MX),K2(MU,MX),K3(MU,MY),K4(MU,MX),D(MU,MU)
      C
      C      THIS SUBROUTINE INITIALIZES THE SYSTEM MATRICES TO ZERO
      C      USING ZOT1
      C
      C
      COMMON/SUBWRIT/ ISUB3NAM
      IF(IISUBNAM.GE.2) WRITE(3,990)
      990 FORMAT(1X,*ZOT*)
      20      MAT1=MX
      MAT2=MY
      CALL ZOT1 (A,MX,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL ZOT1 (C,MX,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      25      MAT1=MY
      MAT2=MX
      CALL ZOT1 (H,MY,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL ZOT1 (G,MY,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      30      MAT2=MU
      CALL ZOT1 (F,MY,MU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT1=MX
      CALL ZOT1 (B,MX,MU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      35      MAT1=MU
      MAT2=MY
      CALL ZOT1 (K1,MU,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL ZOT1 (K2,MU,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      CALL ZOT1 (K3,MU,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      40      CALL ZOT1 (K4,MU,MX,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      MAT2=MU
      CALL ZOT1 (D,MU,MU,
1MX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      RETURN
      END
      ZOT   10
      ZOT   11
      ZOT   12
      ZOT   13
      ZOT   14
      ZOT   15
      ZOT   16
      ZOT   17
      ZOT   18
      ZOT   19
      ZOT   20
      ZOT   21
      ZOT   22
      ZOT   23
      ZOT   24
      ZOT   25
      ZOT   26
      ZOT   27
      ZOT   28
      ZOT   29
      ZOT   30
      ZOT   31
      ZOT   32
      ZOT   33
      ZOT   34
      ZOT   35
      ZOT   36
      ZOT   37
      ZOT   38
      ZOT   39
      ZOT   40
      ZOT   41
      ZOT   42
      ZOT   43
      ZOT   44
      ZOT   45
      ZOT   46
      ZOT   47
      ZOT   48
      ZOT   49
      ZOT   50
      ZOT   51
      ZOT   52

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE ZOT1 73/74 OPT=1 FTN 4.2+75060 01/09/76 14.19.18.

```
      SUBROUTINE ZOT1 (A,N,M,
     1XX,MY,M5,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
      DIMENSION A(MAT1,MAT2)
      COMMON/SUBMPIT/ ISUBNAM
      IF(ISUBNAM.GE.,2) WRITE(3,990)
      340 FORMAT(1X,*ZOT1*)
      DO 10 I=1,N
      DO 10 J=1,M
      A(I,J)=0.0
      10 CONTINUE
      RETURN
      END
```

ZOT1 2
ZOT1 3
ZOT1 4
ZOT1 5
ZOT1 6
ZOT1 7
ZOT1 8
ZOT1 9
ZOT1 10
ZOT1 11
ZOT1 12
ZOT1 13

SUBROUTINE ZTICK (N,NN,DE,I,JJ,ROOTR,RCCTI,ROTR,ROTI,L1,DFE,
 LMX,MY,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
 C
 C THIS SUBROUTINE CONVERTS Z-FLANE TRANSFER FUNCTIONS TO
 C W-PLANE TRANSFER FUNCTIONS
 C
 DIMENSION ROOTR(MX),RCCTI(MX),ROTR(MX),ROTI(MX)
 COMMON/18WRIT/ ISU3NAM
 IF(LGPNAM.GE.2) WRITE(3,990)
 990 FORMAT(1X,*ZTOW*)
 L2=3
 K=1
 IF (I.GT.1.0P.JJ.GT.1) GO TO 1
 DEE=1.0
 1 L1=J
 2 IF (ABS((ROOTR(K)-1.0)+ROTI(K))-1.0GT.10.**(-6)) GO TO 10
 L1=L1+1
 DEE=DEE/2.
 ROOTR(K)=0.0
 ROTI(K)=0.0
 K=K+1
 GO TO 11
 10 IF (ROCTI(K).NE.0.0) GO TO 12
 DEE=DEE/(1.+ROCTR(K))
 ROOTR(K)=-(1.-ROOTR(K))/(1.+RCCTR(K))
 K=K+1
 GO TO 11
 12 A=ROOTR(K)**2
 B=ROCTI(K)**2
 G=(1.+ROCTR(K))**2
 DEE=DEE/(G+B)
 ROOTR(K)=(A+A-1.)/(G+B)
 ROTR(K+1)=ROTR(K)
 ROTI(K)=(2.*ROCTI(K))/(G+B)
 ROTI(K+1)=+ROTI(K)
 K=K+2
 11 IF (K.LE.NN) GO TO 2
 RETURN
 1 DE=DE*DFE
 NSAV=0
 13 IF (ABS((ROTR(K)+1.0)+ROTI(K)).LT.10.**(-6)) GO TO 18
 IF (ABS((ROTR(K)-1.0)+ROTI(K)).GT.10.**(-6)) GO TO 14
 DE=DE*2.
 L2=L2+1
 ROTR(K)=0.
 ROTI(K)=0.
 K=K+1
 GO TO 15
 18 DE=DE*2.
 ROTR(K)=-1000.
 ROTI(K)=0.
 K=K+1
 NSAV=NSAV+1
 GO TO 15
 14 IF (ROTI(K).NE.0.0) GO TO 16
 DE=DE*(1.+ROTR(K))
 ROTR(K)=-(1.-ROTR(K))/(1.+ROTR(K))

SUBROUTINE ZTOW 7/74 OPT=1

FTN 4, 275050 01/09/76 14.22.41.

	K=K+1	ZTOW	59
	GO TO 16	ZTOW	60
60	15 A=ROTR(K)**2	ZTOW	61
	RS=ROTR(K)**2	ZTOW	62
	G=(1.+ROTR(K))**2	ZTOW	63
	DE=DE*(G+R)	ZTOW	64
	ROTR(K)=(A+R-1.)/G+R)	ZTOW	65
65	ROTR(K+1)=ROTR(K)	ZTOW	66
	ROTI(K)=G*ROTR(K)/G+R	ZTOW	67
	ROTI(K+1)=-ROTI(K)	ZTOW	68
	K=K+2	ZTOW	69
70	16 IF (K.LE.N) GO TO 13	ZTOW	70
	LMN=NN-N	ZTOW	71
	DE=DE*(-1.)**LMN	ZTOW	72
	II=1	ZTOW	73
	DO 20 I=1,N	ZTOW	74
	IF (ROTR(II).NE.-1000.) GO TO 22	ZTOW	75
75	K=II	ZTOW	76
	DO 21 L=K,N	ZTOW	77
	ROTR(L)=ROTR(L+1)	ZTOW	78
	ROTI(L)=ROTI(L+1)	ZTOW	79
80	21 CONTINUE	ZTOW	80
	GO TO 20	ZTOW	81
22	II=II+1	ZTOW	82
20	CONTINUE	ZTOW	83
	N=N-NSAV	ZTOW	84
	IF (N.EQ.NN)RETURN	ZTOW	85
85	NI=N+1	ZTOW	86
	N=N-NSAV	ZTOW	87
	IF (N.LE.NI)RETURN	ZTOW	88
	DO 17 I=NI,N	ZTOW	89
	ROTR(I)=I+3	ZTOW	90
90	ROTI(I)=0.	ZTOW	91
	17 CONTINUE	ZTOW	92
	RETURN	ZTOW	93
	END	ZTOW	94

SUBROUTINE COPO 73/74 OPT=1 F7N .. 2475060 01/09/76 14.22.47.

	SUBROUTINE COPO		COPO	2
	C-----		COPO	3
5	C***** SUBROUTINE C C P O PLTS DATA GENERATED BY THE C O N C PROGRAM. THE FOLLOWING TYPES OF PLOTS ARE AVAILABLE.		COPO	4
	C 1. TIME HISTORY		COPO	5
	C 2. TIME HISTORY WITH CSTAR ENVELOPE		COPO	6
	C 3. FREQUENCY RESPONSE		COPO	7
	C 4. POWER SPECTRAL DENSITY		COPO	8
10	C 5. ROOT LOCUS		COPO	9
	C 6. 7-PLANE ROOT LOCUS		COPO	10
	C 7. ROOT CONTOUR		COPO	11
			COPO	12
			COPO	13
			COPO	14
15	C THE MAXIMUM NUMBER OF POINTS FOR ANY ONE PLOT IS 998. TH C MAXIMUM NUMBER OF TIME HISTORY PLOTS IS 21. AN AUTOMATIC C PLOT REQUEST IS GENERATED AT THE END OF TH. JON		COPO	15
	C-----		COPO	16
	C-----		COPO	17
	C-----		COPO	18
			COPO	19
20	EXTERNAL CSTAR		COPO	20
	DOUBLE PRECISION SNAME		COPO	21
	COMMON/CSTAR/CML,CMU,CENVL,CENVU/CSTARP/PTML,FTML,FPNL,PENVU		COPO	22
	C-----		COPO	23
	C ARRAYS IN FOLLOWING DIMENSION STATEMENT DEFINE THE POWER AND CURVE		COPO	24
	C APPROACH CSTAR OPTION ENVELOPES FOR TIME HISTORY PLOTS.		COPO	25
25	C-----		COPO	26
	DIMENSION CML(20),CMU(20),CENVL(20),CENVU(20), X PTML(12),PTMU(23),PENVL(12),PENVU(23)		COPO	27
	X REAL UNCLE(1024),VP(1000),TM,FR,VP(1000), X RIG(10),YLOG(4,9),LOG(9),NOTA,PTITLE(8),COMMENT(12), X PS,RL,TIME(1000),V(25),S10(1000),S20(1000)		COPO	28
30	X INTEGER ORDER(10),LOW,HIGH,LOWY,HIGHY INTEGER YEP		COPO	29
	INTEGER TWO,FOUR,SYSTEM EQUIVALENCE (FREQ,IFRE)		COPO	30
35	INTEGER REMUS,TITLE(8),LAB(4),CNE,NAME(25),ENV,CSCRUV,CSPOA DATA ORDER/1,0,2,5,10,14,12,11,3,7/, X CSCRUV/10HCSTARC / X CSPOA/10HCSTARP / DATA TWO/2/,FOUR/4/,REMUS/1024/, XPC/10HRCO0 /,		COPO	31
40	XFR/10HFR/0 /, XPS/10HSPEC /, XTM/10HTIME /, XRL/10HROLO /,		COPO	32
45	XNPCLCTS/0/,NOTA/10HNOT APP /, XPTITLE/" **** PLEASE READJUS", "T FFN EVER", X"Y FIVE TO ", "TEN PLOT PAGES ****", " ", XCOMMENT/" **** P L E A S E U S E ", " **** ", X" * P R M * F O R A T ", " F D P A ", " P E R ", X" ", " ", " ", " ", " ", " ", X" BIG/2*,18,,25,,20,,18,,25,,18,,20/, X YLOG/,1139,,1761,,2304,,2786,,3617,,3979,,4314,, X .4624,,5185,,5441,,5682,,5911,,6335,,6532,, X .6721,,6902,,7243,,7404,0,,0,,7993,,8129,, X 0,,0,,8673,,8751,0,,0,,9191,,9294,0,,0,, X .9685,,9777,0,,0,, X LOG/,3010,,4771,,6021,,6990,,7782,,8451,,9031,,9542,1/,		COPO	33
50			COPO	34
			COPO	35
			COPO	36
			COPO	37
			COPO	38
			COPO	39
			COPO	40
			COPO	41
			COPO	42
			COPO	43
			COPO	44
			COPO	45
			COPO	46
			COPO	47
			COPO	48
			COPO	49
			COPO	50
			COPO	51
			COPO	52
			COPO	53
			COPO	54
			COPO	55
			COPO	56
			COPO	57
			COPO	58

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE COP0

73/74 OPT=1

FTN 4.2+75060

01/09/76 14.22.47.

```

      XLABEL1/10H MAG=00  /,LABEL1/10H PHI=00  /,LABEL2/10H DUE TO /  COPO  69
      C
      C   SET UP PLOT FACTORS AND READ DATA FOR AUTO PLOT REQUEST  COPO  70
      C
      C   HALF=2./2.54  COPO  71
      C   CM=10./2.54  COPO  72
      C   CALL PLOTS (UNCLF,PEMUS,6)  COPO  73
      C   CALL FACTOR (1.)
      C   RFAD(1,504)SNAME,SURTASK  COPO  74
      C   READ(1,504) IVSN,SNAME,SURTASK  COPO  75
      C 504 FORMAT (15X,2A10,5X,I4)  COPO  76
      C 504 FORMAT (T4,11X,2A10,5X,I4)  COPO  77
      C   WRIT (*,106)  COPO  78
      C   106 FORMAT ("1",10X," PLOTTING HAS BEGUN")  COPO  79
      C
      C   DATA SET # CONTAINS PLOT DATA WRITTEN BY CONTROL PROGRAM  COPO  79
      C   FOR EACH CASE RUN IN THE CONTROL PROGRAM THE FOLLOWING IS WRITTEN ON  COPO  80
      C
      C   RECORD ONE NPLOT=NO. OF PLOTS  COPO  81
      C   RECORD TWO TYPE=TYPE OF PLOT (MUST BE TIME,FREQ,SPEC,ROL)  COPO  82
      C   TITLE=80 CHARACTER PLOT TITLE  COPO  83
      C   SYSTEM=R IMPLIES ROOT LOCUS  COPO  84
      C   MODEL=NOT USED  COPO  85
      C   IDIG=1 INDICATES Z-PLANE ROOT LOCUS  COPO  86
      C   SCAPLT=NCT USED  COPO  87
      C
      C
      C   YEP=0  COPO  88
      C   KLP=0  COPO  89
      C   NCKP=0  COPO  90
      C   NOKP=0  COPO  91
      C   NNKP=0  COPO  92
      C
      9051 CONTINUE  COPO  93
      C   NCKP=NOKP+1  COPO  94
      IF ((YEP,NE.1),AND,(NOKP,EQ.2)) GO TO 99  COPO  95
      IF (NOKP,EQ.2) REWIND 9  COPO  96
      IF (NOKP,EQ.2) GO TO 113  COPO  97
      IF (NCKP,EQ.3) GO TO 99  COPO  98
      REWIND 7  COPO  99
      95 111 CONTINUE  COPO  100
      C   NCO=3  COPO  101
      990 IF (NCO,NE.0) GO TO 108  COPO  102
      IF (NOKP,NE.2) GO TO 1115  COPO  103
      READ(9) NPLOT  COPO  104
      IF (EOF(9),NE.0) GO TO 99  COPO  105
      IF (NOKP,FO.2) GO TO 114  COPO  106
      111 CONTINUE  COPO  107
      RFAD(7)NPLOT  COPO  108
      IF (EOF(7),NE.0) GO TO 8051  COPO  109
      114 CONTINUE  COPO  110
      IF (NOKP,FO.2) NCKP=1  COPO  111
      IF (NOKP,EQ.2) NNKP=KLP  COPO  112
      IF (NOKP,EQ.1) KLP=KLP+1  COPO  113
      1K NCC=NCO+1  COPO  114
      IF (NCO,EQ.NPLOT) NCO=0  COPO  115
      IF (NCKP,NE.2) GO TO 111  COPO  116
      RFAD (9) TYPE,TITLE,SYSTEM,MODEL,IDIG,SCAPLT  COPO  117
      IF (EOF(9),NE.0) GO TO 99  COPO  118
      IF (NOKP,FO.2) GO TO 116  COPO  119

```

SUBROUTINE COPD 23/74 OPT=1 FTN 4.2+75060 01/09/76 14:22:47.
 115 111 READ (7) TYPE,TITLE,SYSTEM,MODEL,LDIG,SCAPLT COPO 115
 IF (IOP(7).NE.0) GO TO 93 COPO 117
 IF((TYPE.EQ.'PL') .AND. (LDIG.EQ.1)) WRITE(91) NPLOT COPO 118
 IF((TYPE.EQ.'PL') .AND. (LDIG.EQ.1)) WRITE(9) TYPE,TITLE,SYSTEM,MODEL, COPO 119
 LDIG,SCAPLT COPO 120
 116 CONTINUE COPO 121
 IF((TYPE.EQ.'TH') .OR. (TYPE.EQ.'FR') .OR. (TYPE.EQ.'PS') .OR. (TYPE.EQ.'RL') .OR. COPO 122
 X.(TYPE.EQ.'DC')) GO TO 1 COPO 123
 WRITE(7,905) COPO 124
 905 FORMAT (10X,'TYPE FIELD DOES NOT CONTAIN VALID OPTION--WILL CONTIN') COPO 125
 XUE SEARCHING FOR RECORD WITH VALTC TYPE FIELD") COPO 126
 GO TO 999 COPO 127
 1 CONTINUE COPO 128
 IF((INCO.EQ.0) .OR. (INCO.EQ.1)) WRITE(3,500) TITLE COPO 129
 500 FORMAT (/5X,9A10) COPO 130
 130 C IS THIS A TIME HISTORY PLOT?
 IF(TYPE.NE.'TH') GO TO 2 COPO 131
 ***** TIME HISTORY PLOT SECTION ***** COPO 132
 131 READ(7,INYU,(NAME(J),J=1,NYU) COPO 133
 H=1 COPO 134
 4 READ(7) TIME(M), (V(K),K=1,NYU) COPO 135
 IF(TIME(M).LT.0.) GO TO 3 COPO 136
 C DATA SET 5 IS A TEMPORARY TIME HISTORY DATA STORAGE COPO 137
 140 C
 WRITE(5) (V(K),K=1,NYU) COPO 138
 MEM+1 COPO 139
 GO TO 4 COPO 140
 3 MEM-1 COPO 141
 145 REWIND 5 COPO 142
 CALL SCALE (TIME,8.5,M,1) COPO 143
 TCM=TIME(M+2) COPO 144
 C----- COPO 145
 150 C----DO LOOP 90 WILL PLOT NYU TIME HISTORY PLOTS----- COPO 146
 DO 90 KK=1,NYU COPO 147
 ENV=C COPO 148
 DO 10 KC=1,M COPO 149
 READ(5) (V(K),K=1,NYU) COPO 150
 10 VP(KC)=V(KK) COPO 151
 C CHECK FOR A CSTAR TYPE TIME HISTORY COPO 152
 C
 IF((NAME(KK).NE.'CSTRU').AND. (NAME(KK).NE.'CSFOA')) ENV=1 COPO 153
 TF(ENV.EQ.1) GO TO 9 COPO 154
 C IF CSTAR OPTION, NORMALIZE DATA BY DIVIDING BY LAST DATA VALUE. COPO 155
 C
 DO 25 KC=1,M COPO 156
 25 VP(KC)=VP(KC)/VP(M) COPO 157
 C REWIND 5 COPO 158
 IF(MOD(KK,4).NE.1) GO TO 31 COPO 159
 IF(KK.EQ.1) GO TO 32 COPO 160
 C SPACE TO NEXT PLOT PAGE COPO 161
 CALL PLOT(28.4950*CM,-2.*CM,-3) COPO 162
 170 C
 COPO 163
 COPO 164
 COPO 165
 COPO 166
 COPO 167
 COPO 168
 COPO 169
 COPO 170
 COPO 171
 COPO 172

SUPPORTING COPO 73/74 OPT=1 FTN 4.2+75060 01/09/76 14.22+7.

```

      52 CALL PLOT (2,*CM,2,*CM,-3)          COPO 173
      CALL FACTOR(HALF)                   COPO 174
      CALL AX90 (0.,0.,"TIME HISTORY---SECONDS",-22,8.5,0.,TIME(M+1), COPO 175
      XTCM)                               COPO 176
      CALL FACTOR(1.)                     COPO 177
      TPIKK,EQ.1) TIME(M+2)=TIME(M+2)*1.27 COPO 178
      YD=-70.                            COPO 179
      YN=13.                            COPO 180
      180 CALL SYMBOL (.5*CM,23,2*CM,.20*CM,TITLE,0.,80) COPO 181
      GO TO 33                           COPO 182
      31 YN=6.                            COPO 183
      YD=YD*.6.                          COPO 184
      33 CALL PLOT (0.,YN*CM,-3)          COPO 185
      CALL SCALE (VP,2.5,M,1)            COPO 186
      C IF ENV NOT EQUAL TO ONE, GENERATE APPROPRIATE CSTAR ENVELOPE COPO 187
      C
      190 IF((NV,EQ.1) GO TO 29           COPO 188
      IF((NAME(KK),EQ.CSCRUV),AND,(VF(M+2).LT..8)) VP(M+2)*.8 COPO 189
      IF((NAME(KK),EQ.CSP0A),AND,(VP(M+2).LT.1.)) VP(M+2)=1. COPO 190
      29 CONTINUE                         COPO 191
      CALL FACTOR(HALF)                   COPO 192
      CALL AX90 (0.,0.,NAME(KK),10,2.5,90.,VP(M+1),VP(M+2)) COPO 193
      VP(M+2)=VP(M+2)*1.27             COPO 194
      CALL FACTOR(1.)                     COPO 195
      C IF ENV NOT EQUAL TO ONE, GENERATE APPROPRIATE CSTAR ENVELOPE COPO 196
      C
      200 IF((AV,EQ.1) GO TO 26           COPO 197
      IF((NAME(KK),NE.CSCRUV) GO TO 28 COPO 198
      CTML(20)=TIME(M+2)                COPO 199
      CENVL(19)=VP(M+1)                 COPO 200
      CENVL(20)=VP(M+2)                 COPO 201
      205 CALL LINE (CTML,CENVL,18,1,0,0) COPO 202
      CTML(20)=CTML(20)                 COPO 203
      CENVL(19)=CENVL(19)               COPO 204
      CENVL(20)=CENVL(20)               COPO 205
      CALL LINE (CTML,CENVL,18,1,0,0) COPO 206
      210 GO TO 26                         COPO 207
      2K PTML(12)=TIME(M+2)              COPO 208
      PENVL(11)=VP(M+1)                 COPO 209
      PENVL(12)=VP(M+2)                 COPO 210
      CALL LINE (PTML,PENVL,10,1,0,0) COPO 211
      PTMU(21)=PTML(12)                 COPO 212
      PENVL(21)=PENVL(11)               COPO 213
      PENVL(22)=PENVL(12)               COPO 214
      215 CALL LINE (PTMU,PENVL,21,1,0,0) COPO 215
      PTMU(23)=PTML(12)                 COPO 216
      PENVL(23)=PENVL(11)               COPO 217
      PENVL(24)=PENVL(12)               COPO 218
      CALL LINE (PTMU,PENVL,21,1,0,0) COPO 219
      220 GO TO 26                         COPO 220
      221 CALL LINE (TIME,VP,M,1,0,0)     COPO 221
      C NOFLOTS KEEPS TRACK OF NUMBER OF PLOTS COPO 222
      NOFLOTS=NOPLOTS+1                 COPO 223
      WRITE(3,502)NAME(KK)              COPO 224
      502 FORMAT (10X,A10," TIME HISTORY PLOT COMPLETED") COPO 225
      90 CONTINUE                         COPO 226
      225 F02 FORMAT (10X,A10,"----- END OF DC LCP 90 -----") COPO 227
      COPO 228
      COPO 229

```

SUBROUTINE DOPD

73/74 OPT=1

FTN 4.2+7500

01/09/78 14:22:47.

```

      C   SPACE TO NEXT PLOT PAGE
      230    CALL PLOT (24,49E0*CH,Y0*CH,-3)
              GO TO 999

      C   IS THIS A FREQUENCY RESPONSE PLCT?
      231    ? TET(TYPE,NE,FRI) GO TO 5
      232
      233
      234
      235
      236
      237
      238
      239
      240
      241
      242
      243
      244
      245
      246
      247
      248
      249
      250
      251
      252
      253
      254
      255
      256
      257
      258
      259
      260
      261
      262
      263
      264
      265
      266
      267
      268
      269
      270
      271
      272
      273
      274
      275
      276
      277
      278
      279
      280
      281
      282
      283
      284
      285
      286

      C-----FREQUENCY RESPONSE AND POWER SPECTRAL DENSITY PLOT SECTION *****
      C
      C   READ(7)NUDF,(NAME(J),J=1,NUDF)
      240    K=1
      C
      C---READ AND CONVERT FREQUENCY RESPONSE DATA-----
      C   CHECK TO SEE IF THERE ARE TWO OR FOUR PLOTS
      C
      244    ? IF(NUDF.EQ.TWO) READ(7)FREQ,VP(K)+VP2(K)
              IF(NUDF.EQ.FOUR) READ(7)FREQ,VP(K),VP2(K),T10(K),S20(K)
              IF(FREQ.EQ.99) GO TO 6
              TIME(K)=ALOG10(FREQ)
              K=K+1
      240    GO TO 7
      241    K=K-1
      242    GO TO 17

      C   IS THIS A POWER SPECTRAL DENSITY PLOT?
      245    ? IF(TYPE,NE,PSI) GO TO 23
      C
      C---THIS SECTION WILL READ AND CONVERT POWER SPECTRAL DENSITY DATA
      C
      C   READ(7)Y,(NAME(J),J=1,2)
      246    YI=16.
      247    YA=-18.
      248    K=1
      249    IF(READ(7)FREQ,VALUE)
              IF(FREQ.EQ.99) GO TO 13
              FREQ=FREQ/6.2832
              TIME(K)=ALOG10(FREQ)
              IF(K.EQ.1) GO TO 35
              IF((TIME(K)).LT.,PTIME) GO TO 13
      250    PTIME=TIME(K)
              IF(VALUE).GT.,34
      251    VP(K)=ALOG10(-VALUE)
              GO TO 44
      252    VP(K)=ALOG10(VALUE)
              GO TO 44
      253    VP(K)=0
      254    IF((VP(K)).LT.,YI) YI=VP(K)
              IF((VP(K)).GT.,YA) YA=VP(K)
      255    K=K+1
      256    GO TO 16
      257    K=K-1
      258
      C-----DETERMINE X AXIS MINIMUM AND INCREMENT
      259    DO 20 J=1,I1
              IF(TIME(1).LT.,FLOAT(I-E-J)) LOW=FLOAT(S-J)
      260
      261
      262
      263
      264
      265
      266

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE CCPD

73/74 CPT=1

FTN 4.2+75060

01/09/76 14.22.47.

```

      IF(TIME(K).GT.FLOAT(J-8))HIGH=FLOAT(J-7)
20  CCNTINUE
      S=LOW
      DO 30 J=1,5
      L=J
      S=S+1
      IF(S.EQ.HIGH) GO TO A
30  CCNTINUE
      8 XLEN=17.
      IF(TYPE,FQ,PS) XLEN=18.
      XCIST=(XLEN/L)*CM
      TIME(K+1)=LOW
      TIME(K+2)=1./XCIST
      NUDI/=1
300   C
C----DC LOOP 101 WILL PLOT ONE PSD OR TWO FREQ. RESPONSE PLOTS-----
C
      5E  DO 101 KK=1,2
      X=2.1*CM
      Y=1.1*CM
      IF(KK.EQ.1) GO TO 221
      X=0.
      Y=-1.1*CM
      IF(TYPE,NE,PS) GO TO 221
      X=0.0
      Y=1.1*CM
    221 IF(KK,NE,21 GO TO 703
      CALL PLOT(X,Y,-3)
      GO TO 11
315   C
C   GENERATE X AXIS (LOG) FOR PSD OR FREQUENCY RESPONSE
C
      703 IF(TYPE,EQ,FR) GO TO 700
      CALL PLOT(CM,0.,-3)
      CALL SYMBOL(.5*CM,25.*2*CM,.20*CM,TITLE,0.,80)
      GO TO 701
      700 CALL PLOT(X,Y,-3)
      CALL SYMBOL (.5*CM,12.2*CM,.20*CM,TITLE,0.,80)
      /01 XN=0.
      YN=0.
      IF(TYPE,FQ,FR)YN=-13.*CM
      LK=L+1
      PL=1.
      DO 40 J=1,LK
      CALL SYMBOL (XN,YN,.65*CM,13.0.,-1)
      IF(J,EG,LK) GO TO 40
      DO 50 J1=1,6
      XP=XN+XDST*LCG(J1)
      CALL SYMBOL (XP,YN,.2*CM,13.0.,-1)
      PL=PL+1.
      XP=XP+.05*CM
      CALL NUMBER (XP,YN-.30*CM,.15*CM,PL,0.,-1)
      50 CALL PLOT (XP,YN,3)
      PL=1.
      40 XN=XN+XDST
      CALL PLOT (0.,YN,2)
      YN=-.2*CM

```

SUBROUTINE ZRPO 73/74 DPT=1 FTN 4,2+75050 01/09/76 14.22+47.

 145 XEM=.4*CM COPO 344
 XNO=LOW COPO 345
 DO 10 J=1,LK COPO 346
 CALL NUMBER (XBM,YN=.70*CM,.25*CM,10.,0.,-1) COPO 347
 CALL NUMBER (XFM,YN=.50*CM,.125*CM,XNO,0.,-1) COPO 348
 XNO=XNO+1, COPO 349
 XPM=XBM*XDIR COPO 350
 10 YFM=XFM*XDIR COPO 351
 C COPO 352
 C IS THIS A PSD PLOT? COPO 353
 IF(TYPE,0,PSD) GO TO 19 COPO 354
 C COPO 355
 356 C GENERATE Y AXIS (CM) FOR FREQUENCY RESPONSE AND PLOT DATA LINE COPO 356
 C COPO 357
 CALL SYMBOL (E,.5*CM,YN=1,10*CM,.25*CM,"FREQUENCY RESPONSE--RADIAN") COPO 358
 X/SEC",0.,31) COPO 359
 401 FORMAT (F10.4) COPO 360
 READ (1,401) DDA
 IF (EOF(1),NE,0) GO TO 902 COPO 362
 IF (DDA,LE,0.) GO TO 902 COPO 363
 VMAX=100. COPO 364
 DO 104 IJK=1,K COPO 365
 IF (VP(IJK) .GT. VMAX) VMAX=VP(IJK) COPO 366
 104 CONTINUE COPO 367
 VMAX=LOAD(TIFIX(VMAX+DDA/2.)) COPO 368
 VP(IK+1)=VMAX-6.0*DDA COPO 369
 VP(K+1)=DDA COPO 370
 470 VMIN=VMAX-11.5*DDA COPO 371
 DO 906 JK=1,K COPO 372
 IF (VP(JK) .LT. VMIN) VP(JK)=VMIN COPO 373
 906 CONTINUE COPO 374
 GO TO 903 COPO 375
 471 H02 CALL SCALE (VP,6.0,K,1) COPO 376
 472 CONTINUE COPO 377
 LAB(1)=LABEL COPO 378
 LAB(3)=LAREL2 COPO 379
 JEP COPO 380
 IF(NUDIE,EQ,1) J=J-2 COPO 381
 LAB(2)=NAME(J+1) COPO 382
 LAB(4)=NAME(J+2) COPO 383
 CALL FACTOR(HALF) COPO 384
 CALL AX90 (0.,0.,LAB,40,6.,90.,VP(K+1),VP(K+2)) COPO 385
 485 VP(K+2)=VP(K+2)*1.27 COPO 386
 CALL FACTOR(1.) COPO 387
 NCPLOTS=NOPLOTS+1 COPO 388
 CALL LINE (TIP,VP,K,1,0,0) COPO 389
 GO TO 12 COPO 390
 490 C GENERATE Y AXIS (LOG) FOR PSD COPO 391
 C COPO 392
 19 CALL SYMBOL (5.25*CM,-1.15*CM,.25*CM,"POWER SPECTRAL DENSITY--CYCL") COPO 394
 X/SEC",0.,34) COPO 395
 DO 70 J=1,37 COPO 396
 IF(YI,LT,FLOAT(1E-J))LOWY=FLOAT(1E-J) COPO 397
 IF(YA,GT,FLOAT(J-18))HIGHY=FLOAT(J-17) COPO 398
 70 CONTINUE COPO 399
 S=LOWY COPO 400

SUBROUTINE C.RPT

73/74 OPT=1

FTN 4.2+75.060

01/09/76 14.22.47.

```

410      DO 80 J=1,10
        LY=J
        S=+1
        IF(S,PG,HIGHY) GO TO 21
        8 CONTINUE
        21 YDIST=(25./LY)*CM
        IND=25./LY
        VP(K+1)=LOWY
        VP(K+2)=1./YDIST
        YN=B.
        LKY=LY+1
        PL=1.
        DO 140 J=1,LKY
        CALL SYMBOL (0.,YN,.3*CM,13,90.,,-1)
        IF(J,EO,LKY) GO TO 140
        140 DO 100 J1=1,9
        IF(IND,LT,.4) GO TO 76
        IF(IND,LT,.6) GO TO 77
        IF(IND,LE,.6) GO TO 78
        IF(IND,LE,.12) GO TO 79
        IF(J1,LT,.5) GO TO 73
        IF(J1,GT,.5) GO TO 79
        DO 110 J2=1,4
        YQ=YN*YDIST*YLOG(J2,J1)
        110 CALL SYMBOL (0.,YQ,.2*CM,13,90.,,-1)
        GO TO 76
        77 IF(J1,GT,.3) GO TO 76
        GO TO 79
        78 IF(J1,GT,.6) GO TO 76
        79 DO 120 JP=1,2
        YQ=YN*YDIST*YLOG(J2,J1)
        120 CALL SYMBOL (0.,YQ,.2*CM,13,90.,,-1)
        76 IF(J1,EO,.9) GO TO 100
        YG=YN*YDIST*LOC(J1)
        CALL SYMBOL (0.,YG,.3*CM,13,90.,,-1)
        IF(IND,LT,.4) GO TO 100
        100 CONTINUE
        PL=PL+1.
        CALL NUMBER (-,25*CM,YQ,.15*CM,PL,0.,,-1)
        100 CONTINUE
        PL=1.
        140 YN=YN+YDIST
        CALL FLOT (0.,0.,2)
        YNO=LOWY
        YEM=.4*CM
        YBH=.2*CM
        DO 170 J=1,LKY
        CALL NUMBER (-,5*CM,YBH,.3*CM,10.,90.,,-1)
        CALL NUMBER (-,8*CM,YEM,.2*CM,YNO,90.,,-1)
        YNO=YNO+1.
        YBH=YBH+YDIST
        170 YEM=YEM+YDIST
        LAB(1)=NAME(1)
        LAB(2)=LABCL2
        LAB(3)=NAME(2)
        CALL SYMBOL (-1.3*CM,10.3*CM,.3*CM,LAB,90.,30)
        GO TO 22
        22 GENERATE SECOND FREQUENCY RESPONSE Y AXIS AND PLOT DATA LINE

```

SUBROUTINE COPO

73/74 OPT=1

FTN 4.2+75 DEC

01/03/74 14:22:47.

```

11 VP2(K+1)=-270.
VP2(K+2)=60.*1.27
410 LAB(1)=LABEL1
LAB(3)=LABEL2
J=2
IF(NUDIE.EQ.1) J=J-2
LAB(2)=NAME(J+1)
LAB(4)=NAME(J+2)
CALL FACTOR(HALF)
CALL AX90(0.,0.,LAB,40,6.,90.,-270.,E0.)
CALL FACTOR(1.)
NOPLOTS=NOPLOTS+1
470 CALL LINE (TIME,VP2,K,1,0,0)
C
C     SPACE TO NEXT PLOT PAGE
12 IF(KK.EQ.2) CALL PLOT (28.4950*CM,0.,-3)
IF(KK.EQ.1) GO TO 101
475 GO TO 101
C
C     PLOT PSD DATA LINE
C
22 CALL LINE (TIME,VP2,K,1,0,0)
NOPLOTS=NOPLOTS+1
476 IF(FPS.EQ.99) GO TO 36
READ(7) FREQ,VALUE,VALUE
GO TO 42
36 CONTINUE
477 C
C     SPACE TO NEXT PLOT PAGE
CALL PLOT (29.4950*CM,0.,-3)
WRITE(3,1113)
478 1113 FORMAT(10X,"P S O  PLOTS COMPLETED")
479 GO TO 98
101 CONTINUE
C
C----- END OF CC LCOP 101 -----
C
480 WRITE(3,1112)
481 1112 FORMAT (10X,"FREQUENCY RESPONSE PLOTS COMPLETED")
C
C     CHECK FOR MORE THAN TWO FREQUENCY RESPONSE PLOTS
C
506 IF(TYPE.EQ.PS1) GO TO 999
IF(NLDC.EQ.THC) GO TO 999
IF(NUDIE.EQ.2) GO TO 999
NUDIE=2
DO 200 JK5=1,K
VP1(JK5)=S10(JK5)
200 VP2(JK5)=S20(JK5)
GO TO 56
507 C
C----- ****ROOT LOCUS AND ROOT CONTOUR SECTION*****-
C
C
510 511
C
C
512 IS THIS A ROOT LOCUS PLOT?
513
514

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE COPO 77/74 OPT=1 FTN 4.2+75060 01/09/74 14.22.47.
 1- 23 IF(TYPE,CO,RL) READ(7)N,M
 METARS(M)
 C
 CHECK FOR A Z-PLANE ROOT LOCUS (IDIG=1)
 IF(NOKP,-0,2) GO TO 1901
 IF(IDIG,NE,1) GO TO 201
 YEP=1
 C-----
 *****Z-PLANE ROOT LOCUS SECTION*****
 C GENERATE HALF CIRCLE AND EACH AXES
 C
 CALL PLOT (17.*CM,2.*CM,-3)
 CALL PLOT (0.,20.*CM,2)
 Z=1.
 Y=20.*CM
 DO 202 J=1,6
 CALL SYMBOL (0.,Y,,2*CM,13.90.,-1)
 CALL NUMBER (1.5*CM,Y-.3*CM,.25*CM,7.90.,1)
 Z=7-.6
 202 Y=Y-.3*CM
 CALL SYMBOL (2.*CM,6.*CM,.3*CM,"2 - PLANE ROOT LOCUS (.1 UNIT/CM X)M, IN.,15)
 Y=11*CM
 CALL PLOT (0.,Y,3)
 X=-12.*CM
 CALL PLOT (X,Y,2)
 Z=1.-2
 DO 203 J=1,3
 CALL PLOT (X,Y,3)
 CALL PLOT (X,Y,2)
 CALL SYMBOL (X,Y,,2*CM,13.90.,-1)
 CALL NUMBER (X,Y+.15*CM,.125*CM,Z,90.,1)
 CALL PLOT (X,Y+.15*CM,3)
 Z=7-.4
 203 X=X+.4*CM
 CALL SYMBOL (-15.0*CM,.5*CM,.24*CM,TITLE(11,90.,80))
 C FOR PLOT LOCUS EACH GAIN INCREMENT HAS A DIFFERENT SYMBOL PLOTTED
 C
 CALL SYMBOL (-15.*CM,3.*CM,.24*CM,"GAIN INCREMENT ORDER---",
 X90.,23)
 Y=9.1*CM
 X=-11.15*CM
 DO 204 J=1,11
 CALL SYMBOL (X,Y,RIGI(J)*CM*1.2,ORDER(J),0.,-1)
 204 Y=Y+.45*CM
 S1=1.31
 DO 205 J=1,100
 S1=S1-.01
 S10(J)=S1
 205 TIME(J)=S0RT(1-S1)(J)**2
 S10(101)=-1.1
 S10(102)=.254
 TIME(101)=-1.2
 TIME(102)=.254
 206

EXECUTIVE 2 100

7/17/74 DPT=1

FTN 4.2+750HD

01/09/74 14.22.47.

```

      CALL PLOT (-12.*CM,0.,,-3)
      CALL LINE (TIME,S10,100+1,0,0)
      J1 =-1,01
      DO 226 J=1,100
      S10(J)=S1
      208 CALL (TIME,S10,100+1,0,0)
      XPT=1.*2.*PI
      YCP=Y-1.*2.*PI
      XMIN=-1.7
      XMAX=1.7
      YMIN=0.
      YMAX=1.7
      GO TO 207
      207
      -----
      **** FOLLOWING SECTION INVOLVES ROOT LOCUS OR ROOT CONTOUR
      READ MAXIMUM AND MINIMUM VALUES FOR Y AXIS AND X AXIS GENERATION
      201 READ (1,900)YMIN,YMAX
      900 FORMAT (2F10.0)
      IF (EOF(1) .NE. 0) GO TO 998
      WRITE(1,903)YMIN,YMAX
      903 FORMAT (10X,"YMIN=",F6.1,EX,"YMAX=",F6.1)

      GENERATE X AXIS
      202 CALL SUBCOL(YMAX,YMIN,25.0,1.,10.,YCRFY)
      YDT=YCRFY
      XMIN=-14.*XDT
      XMAX=4.*XDT
      CALL FACTOR(HALF)
      IF(ICTYPE.EQ.RLY CALL AXIS (0.,0.,"ROOT LOCUS",-10.9.,0.,XMIN,
      XXDT*2.)
      IF(ICTYPE.NE.RLY CALL AXIS (0.,0.,"ROOT CONTOUR",-12.9.,0.,XMIN,
      XXDT*2.)
      CALL FACTOR(1.)
      203 CALL PLOT (14.*CM,0.,,3)
      CALL PLOT (14.*CM,25.*CM,2)
      CALL SYMBOL (CM,25.2*CM,.20*CM,TITLE(1),0.,80)

      IS THIS A ROOT CONTOUR PLOT
      IF(ICTYPE.NE.RLY GO TO 61

      FOR ROOT LOCUS EACH GAIN INCREMENT HAS A DIFFERENT SYMBOL PLOTTED
      204 CALL SYMBOL (CM,26.1*CM,.24*CM,"GAIN INCREMENT ORDER--",0.,23)
      XA=7.10*CM
      YA=25.15*CM
      DC 170 J=1,10
      CALL SYMBOL (XA,YA,.24*CM,ORDERR(J),0.,-1)
      170 XN=XN+.85*CM
      61 CONTINUE
      XA=18.*CM
      YA=25.*CM
      
```

COPD 572
COPD 573
COPD 574
COPD 575
COPD 576
COPD 577
COPD 578
COPD 579
COPD 580
COPD 581
COPD 582
COPD 583
COPD 584
COPD 585
COPD 586
COPD 587
COPD 588
COPD 589
COPD 590
COPD 591
COPD 592
COPD 593
COPD 594
COPD 595
COPD 596
COPD 597
COPD 598
COPD 599
COPD 600
COPD 601
COPD 602
COPD 603
COPD 604
COPD 605
COPD 606
COPD 607
COPD 608
COPD 609
COPD 610
COPD 611
COPD 612
COPD 613
COPD 614
COPD 615
COPD 616
COPD 617
COPD 618
COPD 619
COPD 620
COPD 621
COPD 622
COPD 623
COPD 624
COPD 625
COPD 626
COPD 627
COPD 628

SUBROUTINE COP0

73/74 OPT=1

FTN 4.2 75050

01/09/76 14.22.47.

```

      YF=YCREY*.5,
      YDD=YMIN+25.*YCR*Y
      DO 180 J=1,6
      CALL PLOT (19.*CM,YN,.3)
      CALL SYMBOL (18.*CM,YN,.4*CM,13,90.,-1)
      CALL NUMBER (XN,YN,.26*CM,YDD,0.,1)
      CALL PLOT (18.*CM,YN,.3)
      180 YDD=YDD-Y5
      YCREY=YCREY*.54
      XCT=XCT*2.54
      IF (TYPE,NE,PL) GO TO 62
      640 207 CONTINUE
      C
      C-----DC LOOP 190 WILL READ THE ROOTS WITH N BEING THE NO. OF GAIN INCREME
      C-----AND M BEING THE NO. OF ROOTS TO BE PLOTTED WITHIN EACH GAIN INCREME
      C-----BOTH 7-PLANE AND REGULAR ROOT LOCUS ARE PLOTTED HERE
      C
      DO 190 J=1,N
      DO 190 K=1,M
      IF(K,EG,1) PTF=1.
      IF (NOKP,NE,2) GO TO 1655
      READ (9) I,(TIME(L),VP(L),L=1,I)
      IF (EOF(9),NE,0) GO TO 1900
      IF (NOKP,EO,2) GO TO 115
      1655 CONTINUE
      READ(7I),(TIME(L),VP(L),L=1,I)
      IF (IDIG,EO,1) WRITE(9) I,(TIME(L),VP(L),L=1,I)
      115 CONTINUE
      LL=0
      DO 160 L=1,I
      C
      C     THROW OUT ALL POINTS OUTSIDE X AND Y BOUNDS
      C
      IF(((VP(L)),GT,YMAX),OR,((VP(L)),LT,YMIN),OR,((TIME(L)),GT,XMAX),O
      KR,((TIME(L)),LT,XMIN)) GO TO 160
      IF(((VP(L)),EO,0.),AND,((TIME(L)),EQ,0.)) GO TO 160
      IF((J,NE,1),OR,(K,NE,1)) GO TO 27
      XN=(TIME(L)-XMIN)/XDT
      YN=(VP(L)-YMIN)/YCREY
      IF(IDIG,EO,1) XN=(1.2-VP(L))/XCT
      IF(IDIG,EO,1) YN=(1.+TIME(L))/YCREY
      IF(NOKP,EO,2) XN=-VP(L)/XDT
      IF(NOKP,EO,2) YN=(16.+(TIME(L)-1.)/.005)/2.54
      CALL SYMBOL (XN,YN,.35*CM,1,0.,-1)
      GO TO 160
      675 27 LL=LL+1
      TIME(LL)=TIME(L)
      VP(LL)=VP(L)
      160 CONTINUE
      L=LL
      IF(IDIG,NE,1) GO TO 208
      DO 209 JJ=L,L
      VT=VP(JJ)
      VP(JJ)+TIME(JJ)
      209 TIME(JJ)=-VT
      680
      COP0 629
      COP0 630
      COP0 631
      COP0 632
      COP0 633
      COP0 634
      COP0 635
      COP0 636
      COP0 637
      COP0 638
      COP0 639
      COP0 640
      COP0 641
      COP0 642
      COP0 643
      COP0 644
      COP0 645
      COP0 646
      COP0 647
      COP0 648
      COP0 649
      COP0 650
      COP0 651
      COP0 652
      COP0 653
      COP0 654
      COP0 655
      COP0 656
      COP0 657
      COP0 658
      COP0 659
      COP0 660
      COP0 661
      COP0 662
      COP0 663
      COP0 664
      COP0 665
      COP0 666
      COP0 667
      COP0 668
      COP0 669
      COP0 670
      COP0 671
      COP0 672
      COP0 673
      COP0 674
      COP0 675
      COP0 676
      COP0 677
      COP0 678
      COP0 679
      COP0 680
      COP0 681
      COP0 682
      COP0 683
      COP0 684
      COP0 685

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE: CPO

73/74 OPT=1

FTN 4.2+75050

01/09/76 14.22.47.

```

      94 GO TO 999                               COP0  743
C-----                                     COP0  744
C-----                                     COP0  745
745  ****ROOT CONTOUR PLOT SECTION*****      COP0  746
C-----                                     COP0  747
      62 IFS=1                                COP0  748
      GO TO 67                                COP0  749
      66 IFS=1                                COP0  750
750  67 READ(7), (TIME(L),VP(L),L=1,I)       COP0  751
      IF((TIME(I),EQ.-1.),AND.(VP(I),EQ.-1.)) GO TO 63
      LL=0
      DO 260 L=1,I
      IF((VP(L)),GT,YMAX),OR,((VP(L)),LT,YMIN),OR,((TIME(L)),GT,XMAX),O
      XR,(TIME(L)),LT,XMIN)) GO TO 260
      IF((VP(L)),EQ,0.),AND,((TIME(L)),EQ,0.)) GO TO 260
      XN=(TIME(L)-XMIN)/XDT
      VN=(VP(L)-YMIN)/YCREY
      IF(IFS,GT,0) GO TO 65
      CALL SYMBOL (XN,VN,.30*CM,4,0.,-1)
      GO TO 260
      65 CALL SYMBOL (XN,VN,.10*CM,1,0.,-1)
      260 CONTINUE
      IF(IFS,EQ,0) GO TO 66
      GO TO 67
      63 IDIG=0
      WRITE (7,1119)
1118 FORMAT (20X,"ROOT LOCUS PLOT COMPLETED")
1119 FORMAT (20X,"ROOT CONTOUR PLOT COMPLETED")
770  C-----                                     COP0  771
C-----                                     COP0  772
C-----                                     COP0  773
C-----                                     COP0  774
C-----                                     COP0  775
C-----                                     COP0  776
C-----                                     COP0  777
775  C   SPACE TO NEXT PLOT PAGE
      CALL PLOT (30,4950*CM,0,-3)
      NCPLOTS=NCPLOTS+1
      GO TO 999
      991 WRITE (7,1000)
1000 FORMAT (/10X,"N O P L O T S R E Q U E S T E D")
      GO TO 99
780  C-----                                     COP0  781
C-----                                     COP0  782
C-----                                     COP0  783
      998 WRITE(3,903)
501  FORMAT (10X,"F R R O R - - NO DATA CARD FOR ROOT LOCUS OR ROOT CON
      XTUR PLOT")
      GO TO 999
      79 CONTINUE
      CALL PLOT (0.,0.,993)
790  C-----                                     COP0  791
C-----                                     COP0  792
      791 FORMAT (10X,"F R R O R - - NO DATA CARD FOR ROOT LOCUS OR ROOT CON
      XTUR PLOT")
      GO TO 999
      792 CONTINUE
      CALL PLOT (0.,0.,993)
      793 GENERATE AUTOMATIC PLOT REQUEST
      794 CALL PLOTREC(SNAME,SUBTASK,0,0,0,0,NOTA,NOTA,0,0,C,
      XPTITLE,NCPLOTS,0,IVSN,COMMENT)
      795 XPTITLE,NCPLOTS,0,0,COMMENT)
      796 END

```

SUBROUTINE READ0

73/74 OPT=1

FTN 4.2#7506F

01/09/76 14.23.06.

```

      SUBROUTINE READ0(YMAX,YMIN,XMAX,XMIN,YCREX,YCREY,IOIG,TIME,VP,NOKP READ0    ?
      (1)                                                 READ0    3
      READ0    4
      -----
      D THIS SUBROUTINE READS AND PLOTS THE ZEROS OF THE TRANSFER READ0    5
      FUNCTION FOR A ROOT LOCUS READ0    6
      -----
      D
      10      DIMENSION TIME(1),VP(1) READ0    7
      IF ((NOKP,NE.,2) .OR. (IOIG,NE.,1)) GO TO 1234 READ0    8
      READ (9) I,(TIME(I),VP(I)),L=1,I READ0    9
      IF (EOF(9),NE.,0) GO TO 161 READ0   10
      IF ((NOKP,EO.,2).AND.(IOIG,EO.,1)) GO TO 11 READ0   11
      1234 CONTINUE READ0   12
      READ (7) I,(TIME(I),VP(I),L=1,I) READ0   13
      IF (IOIG,EO.,1) WRITE (9) I,(TIME(I),VP(I),L=1,I) READ0   14
      11 CONTINUE READ0   15
      DO 160 L=1,I READ0   16
      IF ((VP(L),GT,YMAX).OR.((VP(L),LT,YMIN)).OR.((TIME(L),GT,XMAX).OR.((TIME(L),LT,XMIN))) GO TO 160 READ0   17
      XN=(TIME(L)-XMIN)/YCREX-.039 READ0   18
      YN=(VP(L)-YMIN)/YCREY-.049 READ0   19
      IF (IOIG,NE.,1) GO TO 159 READ0   20
      XN=(1.2-VP(L))/YCREX-.039 READ0   21
      YN=(1.+TIME(L))/YCREY-.049 READ0   22
      IF (YCREY,ED,.0127) XN=VP(L)/YCREY -.039 READ0   23
      IF (YCREY,EO,.0127) YN=(1E.+((TIME(L)-1.)/.005)/2.54 -.049 READ0   24
      159 CALL SYM0L(XN,YN,.140 ,54,0,,-1) READ0   25
      160 CONTINUE READ0   26
      161 CONTINUE READ0   27
      RETURN READ0   28
      END READ0   29

```

SUBROUTINE CSTAR 23/74 OPT=1

FTN 4, 247506B

01/09/78 14.23.09.

```

      SUBROUTINE SURSCL(XMAX,XMIN,S,KFAC,DIV,SCALE)
      -----
      THIS SUBROUTINE COMPUTES THE SCALING FACTOR FOR ROOT LOCUS PLOTS
      -----
      DIMENSION X(11)
      REAL FAC(4)
      DATA FAC/1.,2.,5.,10./
      X=XMAX-XMIN
      IF(X.EQ.0.) GO TO 17
      SL=N*S*DIV/10.
      SLEN
      IF(I.GT..E,AND.R.LE.1.) GO TO 401
      IF(R.ALT.S) GO TO 33
      GO TO 41
      17 WRITE(3,201)
      20 FORMAT(" DATUM MAXIMUM AND MINIMUM ARE THE SAME" )
      40 SCALE=.01
      GO TO 5
      401 SCALE=1.
      GO TO 5
      41 DO 4 I=1,499
      DO 4 J=1,3
      IF(P.GE.FAC(J)*10.**(I-1)) GO TO 4
      SCALE=FAC(J)*10.**(I-1)
      GO TO 5
      4 CONTINUE
      GO TO 5
      18 DO 39 J=1,999
      DO 39 I=1,3
      IF(R.LE.FAC(I-J)*10.**(-I+1)) GO TO 39
      SCALE=FAC(5-J)*10.**(-I+1)
      GO TO 5
      39 CONTINUE
      F=A=MOD(XMIN,SCALE)
      19 IF(XMIN.GT.0.) XMIN=XMIN-A
      IF(XMIN.LT.0.) XMIN=XMIN-SCALE-A
      IF(KFAC.GT.0.) GOT 050
      XMIN=XMIN+SCALE*SLEN
      SCALE=SCALE
      50 CONTINUE
      RETURN
      END
      SUBSCL    2
      SUBSCL    3
      SUBSCL    4
      SUBSCL    5
      SUBSCL    6
      SUBSCL    7
      SUBSCL    8
      SUBSCL    9
      SUBSCL   10
      SUBSCL   11
      SUBSCL   12
      SUBSCL   13
      SUBSCL   14
      SUBSCL   15
      SUBSCL   16
      SUBSCL   17
      SUBSCL   18
      SUBSCL   19
      SUBSCL   20
      SUBSCL   21
      SUBSCL   22
      SUBSCL   23
      SUBSCL   24
      SUBSCL   25
      SUBSCL   26
      SUBSCL   27
      SUBSCL   28
      SUBSCL   29
      SUBSCL   30
      SUBSCL   31
      SUBSCL   32
      SUBSCL   33
      SUBSCL   34
      SUBSCL   35
      SUBSCL   36
      SUBSCL   37
      SUBSCL   38
      SUBSCL   39
      SUBSCL   40
      SUBSCL   41
      SUBSCL   42
      SUBSCL   43
      SUBSCL   44
      SUBSCL   45

```

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX 2

NAMELIST CODE

THE CONDITION CODES AND INPUT DATA ARE CONTAINED IN THE NAMELIST CODE AND ARE LISTED BELOW. ALL OF THE CODES AND DATA ARE INITIALIZED TO ZERO AT THE START OF EACH CASE UNLESS THE SAV OPTION IS SET

CONDITION_CODES (INTEGER VARIABLES)

READ,SYSTEM,OUTPUT,MIXED,DIGITL,FRPS,NUMERS,TRESP,NX,NY,NU,NXC,NUC,
ZOH,N1,N2,CONTUR,MULTRT,MODEL,NSCALE,CMAT,NK2,FORM,IPT,IGO,SAV,
IFLAG,READ3

INPUT_DATA (REAL VARIABLES)

DELT,FINALT,IFREC,FFREQ,DELFRQ,M,GAIN1,GAIN2

CONDITION_CODE_DESCRIPTION (INTEGER VARIABLES)

READ	1	DATA MATRICES INPUT THROUGH LOAD SUBROUTINE
	2	DATA MATRICES CONSTRUCTED IN USER WRITTEN MATRIX SUBROUTINE
	3	DATA FROM PREVIOUS CASE ALTERED IN USER WRITTEN CHANGE SUBROUTINE
	4	DATA MATRICES CONSTRUCTED FROM BLOCK DIAGRAM INFORMATION IN CLASS SUBROUTINE
SYSTEM	1	OPEN LOOP SYSTEM ANALYSIS
	2	CLOSED LOOP SYSTEM ANALYSIS
	3	ROOT LOCUS ANALYSIS
OUTPUT	1	$y = Hx$
	2	$y = Hx + G\dot{x}$
	3	$y = Hx + Fu$
	4	$y = Hx + G\dot{x} + Fu$
MIXED	0	NO ACTION
	1	MIXED SYSTEM ANALYSIS (SEE TABLE V.) SYSTEM MATRICES ARE CONSTRUCTED IN A TWO-STEP PROCESS, STEP 1 SPECIFIES OPEN LOOP PLANT (I.E. SPECIFY A,B, C,H,G,F REGARDLESS OF VALUE OF SYSTEM). STEP 2 AUGMENTS PLANT WITH CONTROL SYSTEM DESCRIBED BY BLOCK DIAGRAM USING CLASS. SYSTEM SPECIFIES THE TYPE OF ANALYSIS FOR THE AUGMENTED SYSTEM.
DIGITL	0	CONTINUOUS SYSTEM ANALYSIS
	1	SAMPLED-DATA SYSTEM ANALYSIS
	2	DISCRETE SYSTEM ANALYSIS
		IF DIGITL ≠ 0, DELT SPECIFIES THE SAMPLE PERIOD OF THE DISCRETE OR SAMPLED-DATA SYSTEM.

FRPS	0	NCT APPLICABLE
	1	FREQUENCY RESPONSE CALCULATED FOR EACH TRANSFER FUNTION S-PLANE IF DIGITL = 0 W-PLANE IF DIGITL = 1,2 (DELT REQUIRED)
	-1	S-PLANE FREQUENCY RESPONSES CALCULATED FROM Z-TRANSFER FUNCTIONS WITH DIGITL = 1,2 (DELT REQUIRED)
	2	S-PLANE POWER SPECTRA CALCULATED (DIGITL= 0)
NUMERS	0	NUMERATOR ZEROES OF S- OR Z-TRANSFER FUNCTIONS CALCULATED
	1	NUMERATOR ZEROES NOT CALCULATED
		CONTROL WILL COMPUTE TRANSFER FUNCTION NUMERATOR ZEROES FOR ALL INPUT-OUTPUT PAIRS DEFINED BY THE INPUT AND OUTPUT VECTORS. FOR MIXED SYSTEM ANALYSIS, THE ITHINU AND ITHINY OPTIONS ALLOW UNWANTED TRANSFER FUNCTIONS TO BE ELIMINATED.
TRESP	0	NO ACTION
	N	N TRANSIENT RESPONSES CALCULATED. 'DELT' SPECIFIES INTEGRA- TION STEP SIZE. IF DISC INPUT ROUTINE IS USED, THERE MUST BE N INPUT CARDS AT THE END OF THE DATA CASE GIVING THE INPUT STEP FUNCTION.
NX,NY,NU		DIMENSIONS OF X, Y, AND U VECTORS. IF MIXED = 1, NX, NY, AND NU SPECIFY DIMENSIONS OF THE OPEN LOOP PLANT (STEP 1). STATES ADDED IN STEP 2 OF THE MIXED OPTION AUTOMATI- CALLY INCREMENT NX, NY, AND NU.
NXC,NUC		DIMENSIONS OF STATE AND INPUT VECTORS CORRESPONDING TO THE CONTINUOUS SUBSYSTEM (PLANT) OF A SAMPLED-DATA SYSTEM. THE PLANT MUST BE PARTITIONED IN THE UPPER LEFT POSITION OF THE SYSTEM MATRICES (A,B,H,F,ETC.) NXC ≤ NX , NUC ≤ NU
ZIH		FOR SAMPLED-DATA SYSTEMS, THE NUMBER OF INPUTS TO THE PLANT WHICH ARE OUTPUTS OF ZERO-CRDER-HOLD DEVICES. THSF MUST BE THE FIRST ZIH COMPONENTS OF THE INPUT VECTOR, U.
N1,N2		THE ROOT LOCUS OPTION ALLOWS TWO FEEDBACK CAINS TO BE SPECIFIED. N1 IS THE NUMBER OF ITERATIONS OF THE FIRST VARIABLE (K1,K2) AND N2 IS THE NUMBER OF ITERATIONS OF THE SECCOND VARIABLE (K3,K4). (COMMONLY, N2 = 0). IF N1 > 0, GAIN INCREMENTS ARE ARITHMETIC (0,1,2,3,...) IF N1 < 0, GAIN INCREMNTS ARE GEOMETRIC (0,1,2,4,8,...) Gain increments of second variable are the same as the first;
CONTUR	0	NCT APPLICABLE N2 must be >0.
	1	ROOT CONTOUR OPTION FOR PARAMETER VARIATION STUDIES CONTROL DETERMINES ONLY SYSTEM EIGENVALUES AND RETURNS TO TOP OF PROGRAM FOR NEXT VARIATION. CONTINUES UNTIL CONTUR SFT TO ZERO. (USED WITH IFLAG, READ3, SAV, AND CHANGF)
MULTRT		FOR SAMPLED-DATA SYSTEMS, COMPUTES MULTRT TRANSIENT RESPONSE POINTS FOR EACH SAMPLE PERIOD SO THAT INTERSAMPLE RESPONSE MAY BE INVESTIGATED. ONLY TRANSIENT RESPONSES ARE CALCULATED IF MULTRT IS SET.

MODEL	0	NOT APPLICABLE
	1	MODEL FOLLOWING ON CONSECUTIVE FREQUENCY RESPONSES
NSCALE	0	NOT APPLICABLE
	1	STATE VECTOR TRANSFORMED TO IMPROVE NUMERICAL CONDITIONING IN DETERMINATION OF EIGENVALUES. A MATRIX SCALED BY A DIAGONAL SIMILARITY TRANSFORMATION.
CMAT	0	C MATRIX IS THE IDENTITY MATRIX (C NOT REQUIRED)
	1	C ≠ I (C REQUIRED)
NK2	0	K2 = 0, K4 = 0 (K2, K4 NOT REQUIRED)
	1	K2 ≠ 0 OR K4 ≠ 0 (K2, K4 REQUIRED)
FORM	0	PRINT ONLY FOR OUTPUT
	1	PRINT AND PLOT OUTPUT
	2	PLOT ONLY FOR OUTPUT THE CONTROL PLTTER PROGRAM AUTOMATICALLY SCALES ALL PLOTS EXCEPT ROOT LOCUS PLOTS (WHICH REQUIRE AN EXTRA DATA CARD).
IPT		CODE FOR EXTRA PRINTOUT FOR DEBUGGING
	0	NO EXTRA PRINTING
	1,2	EXTRA PRINTING
I GO		CODE FOR DATA REQUIRED BY CLASS SUBROUTINE
	0	INPUT DATA REQUIRED BY CLASS (TABLE V, STEP 2)
	1	CLASS USES DATA FROM PREVIOUS CASE
SAV	0	DATA MATRICES NOT SAVED
	1	DATA MATRICES SAVED FOR SUBSEQUENT CASES. IF MIXED = 1, CONTROL SAVES MATRICES DEFINED IN STEP 1 (CLASS INPUT DATA, STEP 2, IS NOT DESTROYED AND IS AVAILABLE FOR SUBSEQUENT CASES).
IFLAG	0	ON SUBSEQUENT CASE THE CONDITION CODES AND INPUT DATA ARE ZEROED BEFORE THE CALL TO CARD. CARD READS TITLE, NAMELIST, OUTPUT LABEL, AND INPUT LABEL CARDS
	1	ON SUBSEQUENT CASES THE CONDITION CODES AND INPUT DATA OF THE PRESENT CASE WILL BE USED. CARD READS ONLY A TITLE CARD FOR ALL SUBSEQUENT CASES. (THE OPTION MAY BE CANCELED BY SETTING IFLAG = 0 OR BY END OF DATA DECK).
READ3	0	NO ACTION
	1	ON SUBSEQUENT CASES, READ DEFAULTS TO 3 TO FORCE PROGRAM TO THE CHANGE SUBROUTINE. THE OPTION IS USED WITH IFLAG FOR PARAMETER VARIATION STUDIES.

INPUT DATA DESCRIPTION (REAL VARIABLES)

DELT TIME INCREMENT FOR TRANSIENT RESPONSES AND/OR SAMPLE PERIOD FOR SAMPLED-DATA SYSTEMS, SECCNDS

FINALT FINAL TIME FOR TRANSIENT RESPONSES, SECCNDS

IFREQ,FFREQ,DELFREQ
INITIAL,FINAL, AND INCREMENTAL FREQUENCIES FOR FREQUENCY RESPONSES OR POWER SPECTRA. DELFRQ = 1.1 IS GOOD FOR MOST APPLICATIONS. FREQUENCIES MUST BE SPECIFIED IN (DELFREQ CANNOT EQUAL 1.0)
RADIANSEC. (S-PLANE) EVEN FOR DISCRETE AND SAMPLED-DATA SYSTEMS. IF IFREQ = 0., PROGRAM DEFAULTS TO AN INTERNAL SET OF FREQUENCY POINTS SPACED BETWEEN .1 AND 150. RAD/SEC.
FOR SAMPLED-DATA FREQUENCY RESPONSES CONTROL DEFAULTS IN THE FOLLOWING MANNER,
IF DIGITL#0 AND FRPS #-1 AND IFREQ=0
 IFREQ = TAN (.1*DELT*.5)
 FFREQ = TAN (.9*3.14*.5)
IF DIGITL#0 AND FRPS #-1 AND IFREQ#0
 IFREQ= TAN (IFREQ*DELT*.5)
 FFREQ = TAN (FFREQ*DELT*.5)

M CODE FOR MODIFIED Z-TRANSFER FUNCTION COMPUTATION FOR SAMPLED-DATA SYSTEMS. M IS THE FRACTIONAL SAMPLE PERIOD DELAY AND IS IN THE RANGE $0 \leq M \leq 1$. M = 1. GIVES THE STANDARD Z-TRANSFORM IF THE SIGNAL HAS NO JUMP DISCONTINUITY AT THE SAMPLE INSTANT. M = 0. GIVES THE Z-TRANSFORM WITH A ONE SAMPLE PERIOD DELAY. HOWEVER, NUMERICAL ERRORS LIMIT M TO $M \geq .2$. THEREFORE, IF M=0., THE PROGRAM DEFAULTS TO STANDARD Z-TRANSFORM ANALYSIS. ONLY OPEN LOOP CALCULATIONS (MODIFIED Z-TRANSFER FUNCTIONS AND FREQUENCY RESPONSES) MAY BE PERFORMED WITH THIS OPTION.

GAIN1,GAIN2 ROOT LOCUS GAIN INCREMENTS FOR THE TWO FEEDBACK GAIN VARIABLES ALLOWED WITH THE ROOT LOCUS OPTION. IF NOT SET, PROGRAM DEFAULTS TO GAIN1= 1.0, GAIN2= 1.0.

APPENDIX 3

INPUT AND OUTPUT LISTINGS OF EXAMPLE PROBLEM

A. INPUT LISTING

EXAMPLE PROBLEM LATERAL-DIRECTIONAL AIRPLANE & CONTROL SYSTEM
\$CODE READ=1,MIXED=1,SYSTEM=1,OUTPUT=2,NX=4,NY=5,NU=2,NSCALE=0,SAV=0,
CMAT=1,IPT=0,FRPS=1,IFLAG=0,RFAD3=0,DELT=.05,FINALT=3.,
TRESP=1,FORM=C,IFREQ=.1,FFREQ=20.,DELFRQ=1.11,\$END

ROLLRATE YAWRATE BETA PHI AY
DELTAAAC DELTARC

	4	4	
-5.9	1.7	-15.	
-4	-1.	10.	
-0.004	-1.	-.25	.11
1.			
	4	2	
14.	3.		
-6	-6.		
	.07		

	4	4	
1.	-.02		
-0.02	1.		
		1.	
	5	4	
1.			1.
	1.		
		1.	
-0.348	8.7		-1.
	5	4	

		8.7	
	5	1	
1	1		3
2	4		1
3	8	4	2
4	6		4
5	5		5
			-2.
			1.
			25.
			25.
			.7
			1.
			10.
			2.
1	2	3	4
3	4		5
3	2	1	
1	3		
2	4		
5	5		
2	1		
3	2		
6	3		
			1.

B. OUTPUT LISTING

EXAMPLE PROBLEM LATERAL-STEERAGE AIRPLANE AND CONTROL SYSTEM

CONTINUOUS SYSTEM
MIXED OPTION
OPEN LOOP
LOAD ROUTINE INPUT
TRANSFER FUNCTIONS
FREQUENCY RESPONSES
TRANSIENT RESPONSES

NX = 4	READ = 1	TRESP = 1	CHAT = 1	DELT = .050
NY = 5	SYSTEM = 1	FRPS = 1	NK2 = 0	FINALT = 3.000
NU = 2	MIXED = 1	NUMRS = 0	IFLAG = 0	IFREQ = .100
NYC = 0	OUTPUT = 2	FORM = 0	IGC = 0	DELFRQ = 1.110
NUC = 0	DIGITL = 0	CONTUP = 0	READ3 = 0	FFREQ = 20.000
ZOH = 0	IPT = 0	MULTRT = 0	SAV = 0	GAIN1 = 0.000
N1 = 0	KOUNT = 1	MODEL = 0	NSCALE = 0	GAIN2 = 0.000
N2 = 0				M = 0.000

THE A MATRIX IS

4	4		
-.9900E+01	.1700E+01	-.1500E+02	-0.
-.4100E+00	-.1000E+01	.1000E+02	-0.
-.4000E-02	-.1000E+01	-.2500E+00	.1100E+00
.1000E+01	-0.	-0.	-0.

THE B MATRIX IS

4	2		
.1400E+02	.3000E+01		
-.6000E+00	-.6000E+01		
-0.	.7000E-01		
-0.	-0.		

THE C MATRIX IS

4	4		
.1000E+01	-.2000E-01	-0.	-0.
-.2000E-01	.1000E+01	-0.	-0.
-0.	-0.	.1000E+01	-0.
-0.	-0.	-0.	.1000E+01

THE D MATRIX IS

5	4		
.1000E+01	-0.	-0.	-0.
-0.	.1000E+01	-0.	-0.
-0.	-0.	.1000E+01	-0.
-0.	-0.	-0.	.1000E+01
-.7480E-01	.8700E+01	-0.	-.1000E+01

ORIGINAL PAGE IS
OF POOR QUALITY

THE G MATRIX IS

F 4
-0. -0. -0. -0.
-0. -0. -0. -0.
-0. -0. -0. -0.
-0. -0. -0. -0.
-0. -0. -0. 8700E+01

BLOCK DIAGRAM INPUT PARAMETERS

NO.	TYPE	CONN-C	MOD	PARAM
1	1	-0	-0	-0
2	4	-0	-0	1
3	8	4	-0	2
4	6	-0	-0	4
5	5	-0	-0	5

ITHINY

1 2 3 4 5 -0 -0 -0 -0 -0

ITHINU

3 4 -0 -0 -0 -0 -0

YTCV

1 3
2 4
5

ZTOU

2 1
3 2

YZTOK

6 3

THE FINAL REDUCED SYSTEM IS

THE A MATRIX IS

9	9							
$-5910E+01$	$.1681E+01$	$-.1441E+02$	0.	$.3498E+03$	$.1801E+04$	0.	0.	0.
$-.5182E+00$	$-.9664E+00$	$.9704E+01$	0.	$-.8007E+01$	$-.7714E+04$	0.	0.	0.
$-.4000E-02$	$-.1000E+01$	$-.2E00E+00$	$.1100E+00$	0.	$.4375E+02$	0.	0.	0.
$.1000E+01$	0.	0.	0.	0.	0.	0.	0.	0.
$-.2000E+01$	0.	0.	0.	$-.2500E+02$	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	$.1000E+01$	0.	0.
$-.4333E+02$	$-.4250E+02$	0.	$.5000E+01$	0.	$-.6250E+03$	$-.3500E+02$	$-.1000E+01$	$-.0000E+02$
0.	$.1000E+01$	0.	0.	0.	0.	0.	$.1000E+01$	0.
$.8665E+01$	$.9700E+01$	0.	$-.1000E+01$	0.	0.	0.	0.	$-.1000E+02$

THE B MATRIX IS

9	2
0.	0.
0.	0.
0.	0.
0.	0.
$.1000E+01$	0.
0.	0.
0.	$.1000E+01$
0.	0.
0.	0.

THE H MATRIX IS

5	9							
$.1000E+01$	0.	0.	0.	0.	0.	0.	0.	0.
0.	$.1000E+01$	0.	0.	0.	0.	0.	0.	0.
0.	0.	$.1000E+01$	0.	0.	0.	0.	0.	0.
0.	0.	0.	$.1000E+01$	0.	0.	0.	0.	0.
$.8665E+01$	$.8700E+01$	0.	$-.1000E+01$	0.	0.	0.	0.	0.

THE F MATRIX IS

5	2
0.	0.
0.	0.
0.	0.
0.	0.

THE EIGEN VALUES OF THE SYSTEM ARE

REAL PART IMAGINARY PART

-.38E10438E+02	-.34711706E+02
-.38E10438E+02	.3471170EE+02
-.14860487E+02	-.36751121E+02
-.14461447E+02	.36751121E+02
.71861414E+02	0.
-.13419070E+01	0.
-.16901707E+01	0.
.22354784E+01	0.
.92094333E-02	0.

THE COEFFICIENTS OF THE CHARACTERISTIC EQUATION ORDERED FROM THE LOWEST POWER OF S

.62837635E+06
-.64570F33E+08
-.39355278E+09
-.44325784E+09
-.14167525E+09
-.21935520E+07
.2715E30E+04
.33936257E+04
.7812E7E1E+02
.10000000E+01

THE ROLL RATE/ DELTA AC NUMERATOR GAIN IS .3498E+03

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART IMAGINARY PART

-.40837930E+02	-.48392089E+02
-.40837930E+02	.48392099E+02
.37439375E+02	0.
-.12398567E+01	0.
-.17439393E+01	0.
.48342766E-01	0.
.53467247E-11	0.

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

-.8416620AE+07
.1574C192E+05
-.30410208E+06
-.39531E55E+06
-.14722F65E+06
.10837P23E+04
.47177938E+02
.10000000E+01

ROLL RATE/ DELTA AC FREQUENCY RESPONSE S-PLANE

FREQUENCY RAD/SEC	AMPLITUDE RATIO DB	PHASE ANGLE DEGREES
.1000	-.15460E+02	.06549E+02
.1110	-.148FEE+02	.82507E+02
.1232	-.14283E+02	.78420E+02
.1368	-.13710E+02	.74304E+02
.1518	-.13156E+02	.70173E+02

.1F9E	-.12629E+02	.66047E+02
.1870	-.12131E+02	.61946E+02
.2070	-.11659E+02	.57894E+02
.2300	-.11243E+02	.53915E+02
.2508	-.10457E+02	.50078E+02
.2839	-.10510E+02	.46286E+02
.3152	-.10202E+02	.42681E+02
.3498	-.99311E+01	.39244E+02
.3843	-.96946E+01	.35983E+02
.4210	-.94893E+01	.32925E+02
.4785	-.91118E+01	.30057E+02
.5311	-.81586E+01	.27384E+02
.5895	-.70264E+01	.24902E+02
.6544	-.89120E+01	.22603E+02
.7263	-.88124E+01	.20474E+02
.8062	-.87254E+01	.18504E+02
.8649	-.85488E+01	.16679E+02
.9934	-.85811E+01	.14982E+02
1.1026	-.85211E+01	.13401E+02
1.2239	-.84677E+01	.11921E+02
1.3585	-.84205E+01	.10531E+02
1.4940	-.83768E+01	.92195E+01
1.6739	-.83422E+01	.79763E+01
1.8580	-.83104E+01	.67930E+01
2.0624	-.82828E+01	.56612E+01
2.2892	-.82590E+01	.45732E+01
2.5410	-.82383E+01	.35210E+01
2.8206	-.82203E+01	.24961E+01
3.1308	-.82042E+01	.14899E+01
3.4702	-.81896E+01	.49278E+00
3.8575	-.81757E+01	-.50526E+00
4.2818	-.81619E+01	-.15151E+01
4.7528	-.81477E+01	.25485E+01
5.2756	-.81324E+01	.36183E+01
5.8159	-.81153E+01	.47387E+01
6.5001	-.80955E+01	.59255E+01
7.2151	-.80722E+01	.71969E+01
8.0048	-.80443E+01	.85739E+01
8.8497	-.81105E+01	.10081E+02
9.84676	-.79691E+01	.11750E+02
10.9530	-.79182E+01	.13616E+02
12.1579	-.78556E+01	.15726E+02
13.4952	-.77743E+01	.18139E+02
14.3747	-.76832E+01	.20932E+02
16.6275	-.75667E+01	.24207E+02
18.4565	-.74255E+01	.28099E+02
20.4867	-.72589E+01	.32793E+02

THE YAW RATE / DELTA AC NUMERATOR GAIN IS -.8003E+01

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART	IMAGINARY PART
-.11898556E+03	-.16589430E+02
-.11895556E+03	.16589430E+02
.16E16361E+03	0.
-.10000000E+01	0.
-.19927576E+01	0.
-.61133543E+02	-.65198013E-01
-.61133543E-02	.65198013E-01

ORIGINAL PAGE IS
OF POOR QUALITY

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

```

-.19179647E+05
-.25759641E+04
-.14083768E+08
-.70806513E+08
-.3188814E+07
-.23541571E+04
.74812500E+02
.10000000E+01

```

YAW RATE / DELTA AC FREQUENCY RESPONSE S-PLANE

FREQUENCY RAD/SEC	AMPLITUDE RATIO DR	PHASE ANGLE DEGREES
.1000	-.21300E+02	-.13061E+03
.1110	-.19472E+02	-.12982E+03
.1212	-.17959E+02	-.12999E+03
.1318	-.16674E+02	-.13050E+03
.1518	-.15566E+02	-.13205E+03
.1685	-.14600E+02	-.13364E+03
.1870	-.13754E+02	-.13548E+03
.2076	-.13010E+02	-.13749E+03
.2305	-.12357E+02	-.13963E+03
.2558	-.11783E+02	-.14183E+03
.2839	-.11281E+02	-.14405E+03
.3152	-.10842E+02	-.14626E+03
.3498	-.10459E+02	-.14841E+03
.3843	-.10121E+02	-.15050E+03
.4310	-.99262E+01	-.15249E+03
.4745	-.95656E+01	-.15438E+03
.5311	-.93335E+01	-.15616E+03
.6995	-.91250E+01	-.15785E+03
.8544	-.89360E+01	-.15944E+03
.7263	-.87630E+01	-.16095E+03
.8952	-.86039E+01	-.16240E+03
.9949	-.84573E+01	-.16380E+03
.1034	-.83228E+01	-.16516E+03
1.1026	-.82006E+01	-.16649E+03
1.2239	-.80913E+01	-.16780E+03
1.3585	-.79956E+01	-.16908E+03
1.4040	-.79139E+01	-.17034E+03
1.6739	-.78463E+01	-.17156E+03
1.8580	-.77923E+01	-.17274E+03
2.1624	-.77510E+01	-.17388E+03
2.2892	-.77209E+01	-.17497E+03
2.5410	-.77004E+01	-.17601E+03
2.8207	-.76876E+01	-.17700E+03
3.1706	-.76608E+01	-.17795E+03
3.4752	-.76785E+01	-.17886E+03
3.8575	-.76755E+01	-.17975E+03
4.2818	-.76626E+01	-.18059E+03
4.7528	-.76573E+01	-.18147E+03
5.2755	-.76529E+01	-.18236E+03
5.8559	-.76591E+01	-.18328E+03
6.4001	-.77054E+01	-.18425E+03
7.2151	-.77129E+01	-.18530E+03
8.1088	-.77202E+01	-.18643E+03
8.4897	-.77278E+01	-.18770E+03
9.9676	-.77357E+01	-.18912E+03
10.4530	-.77433E+01	-.19074E+03
12.1579	-.77509E+01	-.19262E+03
13.4952	-.77580E+01	-.19444E+03
14.9797	-.77640E+01	-.19750E+03

1E+0275	-77699E+01	-20070E+03
1E+4565	-77730E+01	-20465E+03
2E+867	-77750E+01	-20955E+03

THE β / DELTA AC NUMERATOR GAIN IS .6604E+01

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART	IMAGINARY PART
-30601356E+03	0.
.31081002E+03	0.
-.88070229E+02	0.
-.9746335E+00	0.
-.20536076E+01	0.
-.84457968E-02	0.

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

.13896621E+06
-.1610948E+08
-.25077612E+08
-.8662900E+07
-.35285104E+05
-.86245561E+02
.10000000E+01

β / DELTA AC FREQUENCY RESPONSE S-PLANE

FREQUENCY RAD/SEC	AMPLITUDE RATIO DB	PHASE ANGLE DEGREES
.1000	.32188E+01	-.23476E+02
.1110	.31589E+01	-.25631E+02
.1232	.24696E+01	-.27927E+02
.1358	.26474E+01	-.30356E+02
.1518	.23389E+01	-.32901E+02
.1695	.20911E+01	-.35542E+02
.1870	.17515E+01	-.38253E+02
.2076	.13684E+01	-.41006E+02
.2305	.94098E+00	-.43769E+02
.2558	.46964E+00	-.46510E+02
.2839	-.44257E-01	-.45198E+02
.3152	-.55833E+00	-.51805E+02
.3498	-.11894E+01	-.54304E+02
.3883	-.14135E+01	-.56648E+02
.4310	-.24666E+01	-.58936E+02
.4785	-.31443E+01	-.61046E+02
.5311	-.38426E+01	-.63021E+02
.5895	-.45578E+01	-.64869E+02
.6544	-.52968E+01	-.66601E+02
.7263	-.60275E+01	-.68235E+02
.8062	-.67765E+01	-.69785E+02
.8949	-.75392E+01	-.71269E+02
.9934	-.83099E+01	-.72701E+02
1.1026	-.90915E+01	-.74091E+02
1.2239	-.98051E+01	-.75443E+02
1.3585	-.10692E+02	-.76758E+02
1.5090	-.11513E+02	-.78032E+02
1.6739	-.12348E+02	-.79257E+02
1.8580	-.13197E+02	-.80425E+02

2.0E24	-1.4059E+02	-81527E+02
2.2992	-1.4933E+02	-82558E+02
2.5410	-1.5816E+02	-83515E+02
2.8526	-1.6709E+02	-84397E+02
3.1306	-1.7602E+02	-85208E+02
3.4712	-1.8505E+02	-85954E+02
3.-876	-1.9418E+02	-86643E+02
4.2818	-2.0321E+02	-87286E+02
4.7528	-2.1235E+02	-87893E+02
5.2716	-2.2139E+02	-88476E+02
5.8554	-2.3045E+02	-89048E+02
6.5011	-2.3957E+02	-89624E+02
7.2151	-2.4855E+02	-90221E+02
8.0048	-2.5777E+02	-90857E+02
8.8847	-2.6678E+02	-91558E+02
9.8675	-2.7542E+02	-92353E+02
10.9530	-2.8444E+02	-93281E+02
12.1579	-2.9383E+02	-94392E+02
13.4952	-3.0277E+02	-95753E+02
14.9737	-3.1187E+02	-97454E+02
16.6275	-3.2050E+02	-99615E+02
18.4555	-3.2925E+02	-10240E+03
20.4867	-3.3796E+02	-10603E+03

THE PHI / DELTA AC NUMERATOR GAIN IS .3498E+03

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART	IMAGINARY PART
-----------	----------------

-4.0437930E+02	-4.8392059E+02
-4.0437930E+02	.48392019E+02
.37439375E+02	0.
-1.2398567E+01	0.
-1.7499391E+01	0.
.49342747E-01	0.

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

.1574E157E+05
-30410210E+06
-1.39E3166E+06
-1.14722565E+06
.10977823E+04
.47177938E+02
.10000000E+01

PHI / DELTA AC FREQUENCY RESPONSE S-PLANE

FREQUENCY RAD/SEC	AMPLITUDE RATIO DB	PHASE ANGLE DEGREES
.1000	.4540TE+01	-34575E+01
.1110	.42252E+01	-74998E+01
.1232	.39043E+01	-11586E+02
.1368	.35707E+01	-15703E+02
.1518	.32177E+01	-19833E+02
.1665	.28393E+01	-23960E+02
.1870	.24304E+01	-28061E+02
.2076	.19868E+01	-32113E+02
.2305	.15057E+01	-36091E+02

.2555	.94536E+00	-.39968E+02
.2839	.42551E+00	-.43721E+02
.3152	-.17309E+00	-.47325E+02
.3499	-.80964E+00	-.50762E+02
.3883	-.14785E+01	-.54013E+02
.4310	-.21797E+01	-.57042E+02
.4785	-.29087E+01	-.59950E+02
.5311	-.36620E+01	-.62622E+02
.5815	-.43624E+01	-.65104E+02
.6544	-.52282E+01	-.67404E+02
.7263	-.60352E+01	-.69532E+02
.8062	-.69546E+01	-.71502E+02
.8949	-.76845E+01	-.73328E+02
.9934	-.85232E+01	-.75024E+02
1.1026	-.93596E+01	-.76606E+02
1.2219	-.10223E+02	-.78085E+02
1.3585	-.11082E+02	-.79475E+02
1.5080	-.11947E+02	-.80787E+02
1.6739	-.12817E+02	-.82030E+02
1.8580	-.13691E+02	-.83214E+02
2.0624	-.14570E+02	-.84345E+02
2.2892	-.15453E+02	-.85433E+02
2.5410	-.16339E+02	-.86486E+02
2.8202	-.17227E+02	-.87510E+02
3.1308	-.18117E+02	-.88517E+02
3.4752	-.19009E+02	-.89514E+02
3.8575	-.19902E+02	-.90512E+02
4.2618	-.20794E+02	-.91522E+02
4.7528	-.21687E+02	-.92555E+02
5.2778	-.22579E+02	-.93625E+02
5.8559	-.23467E+02	-.94745E+02
6.5001	-.24354E+02	-.95932E+02
7.2171	-.25237E+02	-.97204E+02
8.0088	-.26116E+02	-.98580E+02
8.8897	-.26988E+02	-.10009E+03
9.8576	-.27853E+02	-.10176E+03
10.9530	-.28709E+02	-.103625E+03
12.1573	-.29557E+02	-.10573E+03
13.4952	-.30382E+02	-.10815E+03
14.9797	-.31193E+02	-.11094E+03
16.5275	-.31983E+02	-.11421E+03
18.1565	-.32749E+02	-.11811E+03
20.4887	-.33487E+02	-.12280E+03

THE $AY / \Delta AC$ NUMERATOR GAIN IS .29E2F+04

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART	IMAGINARY PART
-.12023271E+02	-.15019465E+02
-.12023271E+02	+.15019465E+02
-.10952181E+02	0.
-.70471725E+00	-.97429378E+00
-.70-71725E+01	+.97429378E+00
.79854692E-02	0.
-.10000000E+02	0.

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

-.46854623E+03
+.8090723E+05

ORIGINAL PAGE IS
OF POOR QUALITY

.71766498E+05
 .48154798E+05
 .11438388E+05
 .10483452E+04
 .44410172E+02
 .10400000E+01

A Y / DELTA AC FREQUENCY RESPONSE S-PLANE

FREQUENCY RAD/SEC	AMPLITUDE RATIO DB	PHASE ANGLE DEGREES
.1000	.72610E+01	-.20545E+03
.1110	.70846E+01	-.20775E+03
.1232	.68748E+01	-.21022E+03
.1368	.66273E+01	-.21263E+03
.1518	.61376E+01	-.21558E+03
.1665	.60013E+01	-.21844E+03
.1870	.56145E+01	-.22138E+03
.2076	.51737E+01	-.22437E+03
.2305	.46758E+01	-.22738E+03
.2558	.41187E+01	-.23036E+03
.2839	.35008E+01	-.23327E+03
.3152	.28210E+01	-.23606E+03
.3498	.20790E+01	-.23858E+03
.3983	.12745E+01	-.24108E+03
.4310	.46787E+00	-.24319E+03
.4785	-.51997E+00	-.24496E+03
.5311	-.13070E+01	-.24629E+03
.5895	-.25468E+01	-.24708E+03
.6544	-.36373E+01	-.24722E+03
.7263	-.47574E+01	-.24658E+03
.8062	-.58846E+01	-.24502E+03
.8949	-.69816E+01	-.24244E+03
.9934	-.80000E+01	-.23886E+03
1.1026	-.88874E+01	-.23442E+03
1.2239	-.96040E+01	-.22947E+03
1.3545	-.10140E+02	-.22444E+03
1.5046	-.10520E+02	-.21976E+03
1.6739	-.10792E+02	-.21566E+03
1.8580	-.11012E+02	-.21222E+03
2.0624	-.11222E+02	-.20934E+03
2.2842	-.11454E+02	-.20687E+03
2.5110	-.11723E+02	-.20461E+03
2.8206	-.12035E+02	-.20238E+03
3.1308	-.12387E+02	-.20002E+03
3.4752	-.12772E+02	-.19741E+03
3.9575	-.13182E+02	-.19442E+03
4.3818	-.13603E+02	-.19098E+03
4.7528	-.14023E+02	-.18703E+03
5.2756	-.14430E+02	-.18251E+03
5.8559	-.14810E+02	-.17742E+03
6.5001	-.15150E+02	-.17168E+03
7.2151	-.15436E+02	-.16529E+03
8.0084	-.15656E+02	-.15827E+03
8.8897	-.15795E+02	-.15059E+03
9.9676	-.15838E+02	-.14227E+03
10.3530	-.15765E+02	-.13331E+03
12.1579	-.15552E+02	-.12375E+03
13.4952	-.15169E+02	-.11365E+03
14.9737	-.14577E+02	-.10317E+03
16.6275	-.13734E+02	-.92598E+02
18.4555	-.12605E+02	-.82409E+02
20.4367	-.11175E+02	-.73240E+02

THE ROLL RATE/DELTA RC NUMERATOR GAIN IS .1801E+04

THE Z-POLES OF THE TRANSFER FUNCTION ARE

```

-.10000000E+01 0.
-.25001000E+02 0.
+.61123953E+01 0.
-.10000000E+02 0.
-.35026731E+01 0.
-.14030276E-12 0.

```

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

```

-.75096080E-09
-.53524306E+04
-.67642014E+04
-.12645209E+04
.16984328E+03
.33390278E+02
.10000000E+01

```

POLL RATE/DELTA RC FREQUENCY RESPONSE S-PLANE

FREQUENCY RAD/SEC	AMPLITUDE RATIO n9	PHASE ANGLE DEGREES
.1000	-1.7819E+03	-2.29895E+02
.1110	-1.7975E+02	-3.1742E+02
.1232	-1.8162E+02	-3.3797E+02
.1368	-1.85383E+02	-3.6047E+02
.1518	-1.8842E+02	-3.8473E+02
.1685	-1.8941E+02	-4.1056E+02
.1870	-1.9284E+02	-4.3767E+02
.2076	-1.9672E+02	-4.6577E+02
.2195	-2.0105E+02	-4.9463E+02
.2568	-2.0585E+02	-5.2371E+02
.2839	-2.1109E+02	-5.5291E+02
.3152	-2.1676E+02	-5.8193E+02
.3498	-2.2283E+02	-6.1051E+02
.3883	-2.2927E+02	-6.3849E+02
.4310	-2.3603E+02	-6.6577E+02
.4785	-2.4310E+02	-6.9232E+02
.5311	-2.5042E+02	-7.1816E+02
.5895	-2.5798E+02	-7.4337E+02
.6554	-2.6574E+02	-7.7605E+02
.7223	-2.7371E+02	-7.9232E+02
.8062	-2.8187E+02	-8.1631E+02
.8949	-2.9022E+02	-8.4010E+02
.9934	-2.9876E+02	-8.6374E+02
1.1026	-3.0752E+02	-8.8719E+02
1.2239	-3.1649E+02	-9.1035E+02
1.3585	-3.2568E+02	-9.3302E+02
1.5080	-3.3507E+02	-9.5492E+02
1.6739	-3.4463E+02	-9.7574E+02
1.4580	-3.5431E+02	-9.9511E+02
2.1624	-3.6404E+02	-1.0127E+03
2.2892	-3.7373E+02	-1.0282E+03
2.5410	-3.8327E+02	-1.0413E+03
2.3206	-3.9253E+02	-1.0520E+03

ORIGINAL PAGE IS
OF POOR QUALITY

3.1318	=.41140E+02	=.10609E+03
3.4752	=.09745E+02	=.10653E+03
3.8575	=.41742E+02	=.10680E+03
4.2819	=.42433E+02	=.10682E+03
4.7528	=.43036E+02	=.10658E+03
5.2796	=.43542E+02	=.10611E+03
5.8359	=.43942E+02	=.10541E+03
6.4001	=.44232E+02	=.10449E+03
7.0151	=.44407E+02	=.10337E+03
8.0488	=.44463E+02	=.10208E+03
8.8897	=.44420E+02	=.10063E+03
9.8676	=.44217E+02	=.99091E+02
10.8530	=.43915E+02	=.97508E+02
12.1579	=.43498E+02	=.95952E+02
13.4952	=.42966E+02	=.94555E+02
14.9797	=.42324E+02	=.93413E+02
16.5275	=.41575E+02	=.92700E+02
18.4555	=.40725E+02	=.92624E+02
20.4867	=.39780E+02	=.93459E+02

THE YAW RATE / DELTA PC NUMERATOR GAIN IS =.3714E+04

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART	IMAGINARY PART
=.10000000E+01	0.
=.15581129E+02	=.24557545E+02
=.15581129E+02	=.24557545E+02
=.10000000E+02	0.
=.67523717E-01	=.16816906E+00
=.67523717E-01	=.16816906E+00

THE COEFFICIENTS OF THE NUMFRATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

.27777778E+03
.14580808E+04
.37954141E+04
.97791700E+04
.12043562E+04
.-2297306E+02
.10000000E+01

YAW RATE / DELTA PC FREQUENCY RESPONSE S-PLANE

FREQUENCY RAD/SEC	AMPLITUDE RATIO DB	PHASE ANGLE DEGREES
.1000	=.19045E+02	=.90021E+02
.1110	=.20524E+02	=.86391E+02
.1272	=.22021E+02	=.81386E+02
.1358	=.23517E+02	=.74458E+02
.1418	=.24878E+02	=.65099E+02
.1495	=.25920E+02	=.53317E+02
.1570	=.26397E+02	=.40252E+02
.1676	=.26274E+02	=.27954E+02
.2305	=.25723E+02	=.18026E+02
.2558	=.24978E+02	=.10859E+02
.2849	=.24205E+02	=.60623E+01
.3152	=.23485E+02	=.30594E+01
.3408	=.22848E+02	=.13526E+01

.3883	-.22302E+02	-.57444E+00
-.4310	-.71841E+02	-.46667E+01
-.745	-.31457E+02	-.85425E+00
-.311	-.21140E+02	-.16181E+01
.895	-.20883E+02	-.26000E+01
-.644	-.20640E+02	-.39855E+01
.7263	-.20526E+02	-.55104E+01
.8052	-.20419E+02	-.72233E+01
.8949	-.20356E+02	-.91100E+01
-.4334	-.20341E+02	-.11153E+02
1.1026	-.20374E+02	-.13332E+02
1.2239	-.20457E+02	-.15620E+02
1.3585	-.20595E+02	-.17992E+02
1.5010	-.20788E+02	-.20377E+02
1.6739	-.21039E+02	-.22753E+02
1.8580	-.21347E+02	-.25057E+02
2.0524	-.21712E+02	-.27344E+02
2.2892	-.22130E+02	-.29229E+02
2.5410	-.22597E+02	-.30986E+02
2.8216	-.23108E+02	-.32463E+02
3.1308	-.23657E+02	-.33620E+02
3.4752	-.24237E+02	-.34423E+02
3.8575	-.24839E+02	-.34851E+02
4.2918	-.25457E+02	-.35885E+02
4.7528	-.26083E+02	-.34518E+02
5.2756	-.26710E+02	-.33748E+02
5.8559	-.27331E+02	-.32580E+02
6.5001	-.27941E+02	-.31025E+02
7.2151	-.28537E+02	-.29099E+02
8.0088	-.29104E+02	-.26822E+02
8.8997	-.29649E+02	-.24217E+02
9.8676	-.30168E+02	-.21307E+02
10.9530	-.30657E+02	-.18114E+02
12.1579	-.31115E+02	-.14655E+02
13.4952	-.31540E+02	-.10941E+02
14.9797	-.31927E+02	-.69774E+01
16.6275	-.32265E+02	-.27799E+01
18.4566	-.32535E+02	-.16525E+01
20.4857	-.32705E+02	-.62035E+01

THE BETA / DELTA PC NUMERATOR GAIN IS .4375E+02

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART	IMAGINARY PART
-.10000000E+01	0.
-.95275345E+02	0.
-.15664379E+02	-.24652880E+02
-.15664379E+02	.24652880E+02
-.10000000E+02	0.
.38894055E-02	0.

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

-.28297033E+04
.72429157E+06
.83510518E+06
.11267429E+06
.48169837E+04
.12760321E+03
.10000000E+01

BETA	/DELTA RC FREQUENCY RESPONSE S-PLANE	
FREQUENCY RAD/SEC	AMPLITUDE RATIO DB	PHASE ANGLE DEGREES
.1000	-74410E+01	-20030E+03
.1110	-75992E+01	-21044E+03
.1232	-77884E+01	-21278E+03
.1358	-90119E+01	-21523E+03
.1481	-82734E+01	-21794E+03
.1605	-85758E+01	-22010E+03
.1730	-83220E+01	-22378E+03
.1870	-93139E+01	-22644E+03
.2076	-37326E+01	-22997E+03
.2305	-10238E+02	-23314E+03
.2839	-10770E+02	-23633E+03
.3152	-11346E+02	-23951E+03
.3498	-11964E+02	-24265E+03
.3883	-12622E+02	-24576E+03
.4310	-13115E+02	-24882E+03
.4785	-14042E+02	-25182E+03
.5311	-14799E+02	-25478E+03
.5895	-15586E+02	-25770E+03
.6544	-16401E+02	-26061E+03
.7243	-17243E+02	-26350E+03
.8042	-18116E+02	-26640E+03
.8949	-19020E+02	-26932E+03
.9934	-19959E+02	.87744E+02
1.026	-22938E+02	.84785E+02
1.2339	-21960E+02	.81819E+02
1.3545	-23029E+02	.78871E+02
1.4050	-24150E+02	.75973E+02
1.4779	-25323E+02	.73167E+02
1.4680	-26562E+02	.70502E+02
2.0624	-27834E+02	.68023E+02
2.2342	-29167E+02	.65796E+02
2.5410	-30547E+02	.63855E+02
2.8205	-31970E+02	.62250E+02
3.1308	-33429E+02	.61017E+02
3.4702	-34917E+02	.60183E+02
3.4575	-36428E+02	.59786E+02
4.2818	-37957E+02	.59827E+02
4.7528	-39404E+02	.60321E+02
5.2756	-41016E+02	.61270E+02
5.8559	-42541E+02	.62671E+02
6.5001	-44053E+02	.64516E+02
7.2151	-45545E+02	.66791E+02
8.0068	-47017E+02	.69479E+02
8.8437	-48460E+02	.72562E+02
9.8276	-49874E+02	.76020E+02
10.9530	-51256E+02	.79836E+02
12.1579	-52604E+02	.83999E+02
13.4332	-53915E+02	.88504E+02
14.9797	-55318E+02	-26665E+03
16.6275	-56398E+02	-26146E+03
18.4565	-57538E+02	-25995E+03
20.4657	-58673E+02	-25020E+03

THE PHI /DELTA RC NUMERATOR GAIN IS .1801E+04

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART IMAGINARY PART

.+10000000E+01	0.
.61123957E+01	0.
-.2E000000E+02	0.
-.35026731E+01	0.
-.10000000E+02	0.

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

-.53524306E+04
-.67542014E+04
-.12645208E+04
+.16964028E+03
.33390279E+02
.10000000E+01

PHI / DELTA RC FREQUENCY RESPONSE S-PLANE

FREQUENCY RAD/SEC	AMPLITUDE RATIO DB	PHASE ANGLE DEGREES
.1000	.71805E+01	-.11993E+03
.1110	.11181E+01	-.12175E+03
.1232	.24909E-01	-.12380E+03
.1368	-.11026E+01	-.12605E+03
.1518	-.22677E+01	-.12848E+03
.1675	-.34736E+01	-.13106E+03
.1837	-.47226E+01	-.13377E+03
.2005	-.50167E+01	-.13658E+03
.2175	-.73564E+01	-.13946E+03
.2558	-.97427E+01	-.14238E+03
.2839	-.10177E+02	-.14530E+03
.3122	-.11647E+02	-.14820E+03
.3498	-.13160E+02	-.15106E+03
.3883	-.14711E+02	-.15356E+03
.4310	-.16294E+02	-.15658E+03
.4785	-.17906E+02	-.15924E+03
.5311	-.19545E+02	-.16182E+03
.5895	-.21207E+02	-.16434E+03
.6546	-.22891E+02	-.16681E+03
.7253	-.24594E+02	-.16924E+03
.8062	-.26316E+02	-.17164E+03
.8949	-.28057E+02	-.17402E+03
.9934	-.29819E+02	-.17638E+03
1.1026	-.31601E+02	-.17873E+03
1.2239	-.33404E+02	-.18102E+03
1.3585	-.35230E+02	-.18328E+03
1.5080	-.37075E+02	-.18547E+03
1.6739	-.38938E+02	-.18755E+03
1.8580	-.40812E+02	-.18949E+03
2.0624	-.42692E+02	-.19125E+03
2.2892	-.44567E+02	-.19280E+03
2.5410	-.46427E+02	-.19411E+03
2.8206	-.4820E+02	-.19518E+03
3.1308	-.50051E+02	-.19598E+03
3.4752	-.51793E+02	-.19651E+03
3.4575	-.53468E+02	-.19678E+03
4.2318	-.55065E+02	-.19680E+03
4.7528	-.56575E+02	-.19656E+03
5.2756	-.57987E+02	-.19609E+03
5.5699	-.59294E+02	-.19539E+03
6.1001	-.60491E+02	-.19447E+03

ORIGINAL PAGE IS
OF POOR QUALITY

7.2151	-61571E+02	-19335E+03
8.0038	-52534E+02	-19206E+03
8.8897	-63375E+02	-19061E+03
9.8576	-64111E+02	-18907E+03
10.9350	-64706E+02	-18749E+03
12.1379	-65195E+02	-18594E+03
13.4972	-65570E+02	-18453E+03
14.3797	-65834E+02	-18339E+03
16.6275	-65992E+02	-18268E+03
18.4565	-66048E+02	-18260E+03
20.4867	-66009E+02	-18344E+03

THE AY /DELTA RC NUMERATOR GAIN IS -.1671E+05

THE ZEROS OF THE TRANSFER FUNCTION ARE

REAL PART	IMAGINARY PART
-----------	----------------

-10000000E+01	0.
-19853147E+02	-36396151E+02
-19633147E+02	.36396151E+02
-42251286E+00	0.
.54473431E-02	0.
-10000000E+02	0.

THE COEFFICIENTS OF THE NUMERATOR POLYNOMIAL ORDERED FROM THE LOWEST POWER OF S

-.79567357E+02
.70404524E+04
.25116682E+05
.20111111E+05
.71742719E+04
.50723320E+02
.10000000E+01

AY /DELTA RC FREQUENCY RESPONSE S-PLANE

FREQU-NCY RAD/SEC	AMPLITUDE RATIO DB	PHASE ANGLE DEGREES
.1000	.43855E+01	-.14232E+02
.1110	.42791E+01	-.15074E+02
.1232	.41516E+01	-.15968E+02
.1368	.40075E+01	-.16899E+02
.1518	.38405E+01	-.17846E+02
.1685	.36508E+01	-.18785E+02
.1870	.34403E+01	-.19692E+02
.2076	.32102E+01	-.20542E+02
.2305	.29630E+01	-.21310E+02
.2558	.277023E+01	-.21978E+02
.2839	.24328E+01	-.22533E+02
.3132	.21593E+01	-.22969E+02
.3443	.19873E+01	-.23293E+02
.3443	.16215E+01	-.23520E+02
.3110	.13667E+01	-.23674E+02
.4785	.11258E+01	-.23787E+02
.5311	.90055E+00	-.23899E+02
.6895	.69117E+00	-.24049E+02
.6564	.49611E+00	-.24280E+02
.7263	.31211E+00	-.24629E+02
.8382	.13440E+00	-.25129E+02

.8949	-1.3125E+01	-2.25802E+02
.9934	-2.2755E+00	-2.26665E+02
1.1026	-1.42652E+00	-2.2717E+02
1.2270	-1.64773E+00	-2.2947E+02
1.3385	-1.43947E+00	-3.0331E+02
1.3080	-1.11848E+01	-3.1830E+02
1.5739	-1.01118E+01	-3.3400E+02
1.4580	-1.18822E+01	-3.4946E+02
2.0624	-1.22972E+01	-3.6532E+02
2.2892	-1.27561E+01	-3.7942E+02
2.4110	-1.32561E+01	-3.9241E+02
2.4216	-1.37332E+01	-4.0383E+02
3.1308	-1.43621E+01	-4.1246E+02
3.4772	-1.15665E+01	-4.1838E+02
3.8376	-1.55699E+01	-4.2134E+02
4.2418	-1.51951E+01	-4.2119E+02
4.7528	-1.69250E+01	-4.1787E+02
5.2756	-1.75275E+01	-4.1140E+02
5.4559	-1.90718E+01	-4.0187E+02
6.5001	-1.46761E+01	-3.947E+02
7.2141	-1.22604E+01	-3.7447E+02
8.0088	-1.38285E+01	-3.5721E+02
8.8897	-1.11353E+02	-3.3810E+02
9.5676	-1.13857E+02	-3.1762E+02
10.3530	-1.11332E+02	-2.9631E+02
12.1579	-1.11778E+02	-2.7476E+02
13.4982	-1.21944E+02	-2.5362E+02
14.9797	-1.25965E+02	-2.3361E+02
16.6275	-1.2975E+02	-2.1552E+02
18.4565	-1.13340E+02	-2.0031E+02
20.4867	-1.13697E+02	-1.8914E+02

THE TRANSITION MATRIX 11 TERMS

9	9	9	9	9	9	9	9	9	9
-1.3585E+00	-1.730E+00	-1.4430E+00	.9217E-01	.7458E+01	.6557E+02	.1036E+01	-1.2057E-01	.7287E+00	
.1734E+01	.7963E+01	.3737E+00	-1.2360E+00	.6289E+01	-1.2055E+03	-1.2887E+01	.5190E-01	-1.1947E+01	
-1.4516E-01	-1.1940E-01	.5770E+00	.1148E-01	-1.452E+00	.7171E+01	.8679E-01	-1.1306E-02	.1717E-01	
.2428E+01	-1.9884E-02	-1.12E+01	.1001E+01	.204EE+00	.1766E+01	.2046E-01	-1.298E-03	.1655E-01	
-1.1920E-01	.1628E-01	.1168E-01	-1.2052E-02	.3643E-01	-1.2376E+01	-1.3029E-01	.4516E-03	-1.1638E-01	
-1.2322E-01	-1.2960E-01	.1553E-02	.7461E-02	-1.1725E+00	.1706E+01	.2922E-01	-1.7794E-03	.7666E-01	
-1.6350E+00	-1.1147E+01	.5618E-01	.1156E+00	-1.4957E+01	.5005E+02	.6899E+00	-1.2824E-01	.8943E+00	
.2423E-01	.7457E-01	.1160E-01	-1.3172E-02	.6630E-01	-1.4648E+01	-1.5100E-01	.9519E+00	-1.2501E-01	
.3349E+00	.4729E+00	-1.2312E-01	-1.5424E-01	.1974E+01	-1.2226E+02	-1.2441E+00	.3191E-02	.4903E+00	

TIME	ROLL RATE	YAW RATE	PETA	PHI	AY	DELTA AC	DELTA RC
0.	0.	0.	0.	0.	0.	-0.	.100E+01
.500E-01	.209E-01	-1.525E-01	.132E-02	.293E-03	-1.276E+00	-0.	.100E+01
.100E+00	.129E+00	-.415E+00	.150E-01	.346E-02	-1.250E+01	-0.	.100E+01
.150E+00	.619E+00	-1.219E+01	.901E-01	.191E-01	-1.137E+02	-0.	.100E+01
.200E+00	.303E+01	-.1105E+02	.468E+00	.950E-01	-1.692E+02	-0.	.100E+01
.250E+00	.149E+02	-1.542E+02	.234E+01	.466E+00	-1.343E+03	-0.	.100E+01
.300E+00	.733E+02	-1.267E+03	.115E+02	.230E+01	-1.169E+04	-0.	.100E+01
.350E+00	.361E+03	-1.131E+04	.568E+02	.113E+02	-8.30E+04	-0.	.100E+01
.400E+00	.177E+04	-.645E+04	.280E+03	.557E+02	-4.40E+05	-0.	.100E+01
.450E+00	.873E+04	-1.175E+05	.137E+04	.274E+03	-2.01E+06	-0.	.100E+01
.500E+00	.429E+05	-.156E+06	.676E+04	.135E+04	-9.88E+06	-0.	.100E+01
.550E+00	.211E+06	-1.78E+06	.373E+05	.653E+04	-4.86E+07	-0.	.100E+01
.600E+00	.104E+07	-3.78E+07	.164E+06	.326E+05	-2.39E+08	-0.	.100E+01
.650E+00	.511E+07	-1.98E+08	.805E+06	.160E+06	-1.18E+09	-0.	.100E+01
.700E+00	.251E+08	-3.14E+08	.396E+07	.789E+06	-5.78E+09	-0.	.100E+01
.750E+00	.124E+09	-4.50E+09	.195E+08	.388E+07	-2.84E+10	-0.	.100E+01
.800E+00	.608E+09	-2.21E+10	.958E+08	.191E+08	-1.40E+11	-0.	.100E+01
.850E+00	.299E+10	-1.09E+11	.471E+09	.939E+08	-6.88E+11	-0.	.100E+01

.900E+00	.147E+11	-.575E+11	.232E+10	.452E+09	-.339E+12	-0.	.100E+01
.950E+00	.724E+11	-.263E+12	.114E+11	.227E+10	-.167E+13	-0.	.100E+01
.100E+01	.386E+12	-.129E+13	.561E+11	.112E+11	-.819E+13	-0.	.100E+01
.105E+01	.175E+13	-.537E+13	.276E+12	.550E+11	-.403E+14	-0.	.100E+01
.110E+01	.981E+13	-.313E+14	.176E+13	.270E+12	-.198E+15	-0.	.100E+01
.115E+01	.424E+14	-.154E+15	.667E+13	.133E+13	-.975E+15	-0.	.100E+01
.120E+01	.208E+15	-.758E+15	.328E+14	.654E+13	-.480E+16	-0.	.100E+01
.125E+01	.102E+16	-.373E+16	.161E+15	.322E+14	-.236E+17	-0.	.100E+01
.130E+01	.504E+16	-.183E+17	.794E+15	.158E+15	-.116E+18	-0.	.100E+01
.135E+01	.248E+17	-.902E+17	.391E+16	.778E+15	-.571E+18	-0.	.100E+01
.140E+01	.122E+18	-.444E+18	.192E+17	.383E+16	-.281E+19	-0.	.100E+01
.145E+01	.600E+18	-.219E+19	.945E+17	.188E+17	-.138E+20	-0.	.100E+01
.150E+01	.295E+19	-.107E+20	.465E+18	.92EE+17	-.679E+20	-0.	.100E+01
.155E+01	.145E+20	-.528E+20	.229E+19	.456E+18	-.334E+21	-0.	.100E+01
.160E+01	.714E+20	-.260E+21	.112E+20	.224E+19	-.164E+22	-0.	.100E+01
.165E+01	.351E+21	-.126E+22	.553E+20	.110E+20	-.808E+22	-0.	.100E+01
.170E+01	.173E+22	-.628E+22	.272E+21	.542E+20	-.398E+23	-0.	.100E+01
.175E+01	.850E+22	-.303E+23	.134E+22	.257E+21	-.196E+24	-0.	.100E+01
.180E+01	.418E+23	-.152E+24	.659E+22	.131E+22	-.962E+24	-0.	.100E+01
.185E+01	.206E+24	-.748E+24	.324E+23	.645E+22	-.473E+25	-0.	.100E+01
.190E+01	.101E+25	-.348E+25	.159E+24	.317E+23	-.233E+26	-0.	.100E+01
.195E+01	.497E+25	-.191E+26	.794E+24	.156E+24	-.114E+27	-0.	.100E+01
.200E+01	.248E+26	-.830E+26	.385E+25	.768E+24	-.563E+27	-0.	.100E+01
.205E+01	.120E+27	-.438E+27	.190E+26	.778E+25	-.277E+28	-0.	.100E+01
.210E+01	.592E+27	-.215E+28	.913E+26	.146E+26	-.136E+29	-0.	.100E+01
.215E+01	.291E+28	-.106E+29	.419E+27	.914E+26	-.670E+29	-0.	.100E+01
.220E+01	.147E+29	-.521E+29	.236E+28	.450E+27	-.330E+30	-0.	.100E+01
.225E+01	.705E+29	-.256E+30	.111E+29	.221E+28	-.162E+31	-0.	.100E+01
.230E+01	.347E+30	-.126E+31	.546E+29	.109E+29	-.798E+31	-0.	.100E+01
.235E+01	.170E+31	-.620E+31	.269E+30	.535E+29	-.392E+32	-0.	.100E+01
.240E+01	.839E+31	-.305E+32	.172E+31	.763E+30	-.193E+33	-0.	.100E+01
.245E+01	.412E+32	-.150E+33	.650E+31	.129E+31	-.949E+33	-0.	.100E+01
.250E+01	.207E+33	-.733E+33	.320E+32	.637E+31	-.467E+34	-0.	.100E+01
.255E+01	.999E+33	-.363E+34	.157E+33	.313E+32	-.230E+35	-0.	.100E+01
.260E+01	.491E+34	-.179E+35	.773E+33	.154E+33	-.113E+36	-0.	.100E+01
.265E+01	.241E+35	-.874E+35	.380E+34	.758E+33	-.556E+36	-0.	.100E+01
.270E+01	.119E+36	-.432E+36	.187E+35	.373E+34	-.273E+37	-0.	.100E+01
.275E+01	.584E+36	-.213E+37	.920E+35	.193E+35	-.134E+38	-0.	.100E+01
.280E+01	.287E+37	-.105E+38	.453E+36	.902E+35	-.661E+38	-0.	.100E+01
.285E+01	.141E+38	-.514E+38	.223E+37	.444E+36	-.325E+39	-0.	.100E+01
.290E+01	.695E+38	-.253E+39	.110E+38	.218E+37	-.160E+40	-0.	.100E+01
.295E+01	.342E+39	-.124E+40	.539E+38	.107E+38	-.787E+40	-0.	.100E+01
.300E+01	.168E+40	-.612E+40	.265E+39	.528E+38	-.387E+41	-0.	.100E+01

TABLE I. CONTROL PROGRAM ANALYSIS OPTIONS

CONTINUOUS SYSTEMS

OPEN AND CLOSED LOOP SYSTEMS	
TRANSFER FUNCTIONS	(S-PLANE)
FREQUENCY RESPONSES	(S-PLANE)
POWER SPECTRA	(S-PLANE)
TRANSIENT RESPONSES	(S-PLANE)
ROOT LOCUS	(S-PLANE)

DISCRETE SYSTEMS

OPEN AND CLOSED LOOP SYSTEMS	
TRANSFER FUNCTIONS	(Z- OR W-PLANE)
FREQUENCY RESPONSES	(W- OR S-PLANE)
TRANSIENT RESPONSES	
ROOT LOCUS	(Z-PLANE)

SAMPLED-DATA SYSTEMS

OPEN LOOP SYSTEMS	
STANDARD OR MODIFIED Z-TRANSFER FUNCTIONS	(Z- OR W-PLANE)
FREQUENCY RESPONSES	(W- OR S-PLANE)
TRANSIENT RESPONSES	
OPEN AND CLOSED-LOOP SYSTEMS	
STANDARD Z-TRANSFER FUNCTIONS	(Z- OR W-PLANE)
FREQUENCY RESPONSES	(W- OR S-PLANE)
TRANSIENT RESPONSES	
ROOT LOCUS	(Z-PLANE)

SYSTEMS MAY BE DEFINED BY,

1. MATRICES
 2. PARAMETERS
 3. BLOCK DIAGRAM
 4. COMBINATION OF (1.,2.) AND (3.)
- | | |
|----------|--|
| (LOAD) | |
| (MATRIX) | |
| (CLASS) | |
| (MIXED) | |

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE II CONTROL PROGRAM JOB CONTROL LANGUAGE AS OF 11-1-74

JOB CONTROL CARDS FOR
STANDARD CONTROL RUN
WITHOUT SOURCE DECK
WITHOUT PLOT TAPE

```
JOBN,CM7200,T4777.  
ATTACH(JWELIB, ID=JWE, PW=ADDY, MR=1)  
ATTACH(OVRLY, ID=JWE)  
SEGLOAD(I=OVRLY)  
LOAD(JWELIB)  
EXECUTE.  
789 PUNCH  
    DATA  
6789 PUNCH      (YELLOW CARD)
```

JOB CONTROL CARDS FOR
STANDARD CONTROL RUN
WITH SOURCE DECK
WITHOUT PLOT TAPE

```
JOBN,CM7200,T4777.  
ATTACH(JWELIB, ID=JWE, PW=ADDY, MR=1)  
FTN(LR=OUTPUT)  
REWIND(LGO)  
COPYL(JWELIB,LGO,JWE)  
ATTACH(OVRLY, ID=JWE)  
SEGLOAD(I=OVRLY)  
LOAD(JWE)  
EXECUTE.  
789 PUNCH  
    SOURCE DECKS  
789 PUNCH  
    DATA  
6789 PUNCH      (YELLOW CARD)
```

JOB CONTROL CARDS FOR
STANDARD CONTROL RUN
WITHOUT SOURCE DECK
WITH PLOT TAPE

```
JOBN,CM72000,T4777,NT1.  
ATTACH(JWELIB, ID=JWE, PW=ADDY, MR=1)  
LABEL (CARD) SEE DESCRIPTION  
ATTACH(OVRLY, ID=JWE)  
SEGLOAD(I=OVRLY)  
LOAD(JWELIB)  
EXECUTE.  
EXIT.  
BKSP(INPUT)  
ATTACH(PLOTRC, ID=JWE, PW=ADDY, MR=1)  
LOAD(PLOTRC)  
REDUCE.  
EXECUTE.  
789 PUNCH  
DATA  
PLOT CARDS (SEE DESCRIPTION )  
6789 PUNCH (YELLOW CARD)
```

JOB CONTROL CARDS FOR
STANDARD CONTROL RUN
WITH SOURCE DECK
WITH PLOT TAPE

```
JOBN,CM72000,T4777,NT1.  
ATTACH(JWELIB, ID=JWE, PW=ADDY, MR=1)  
LABEL (CARD) SEE DESCRIPTION  
FTN(LR=OUTPUT)  
REWIND(LGO)  
COPYL(JWELIB,LGO,JWE)  
ATTACH(OVRLY, ID=JWE)  
SEGLOAD(I=OVRLY)  
LOAD(JWE)  
EXECUTE.  
EXIT.  
BKSP(INPUT)  
ATTACH(PLOTRC, ID=JWE, PW=ADDY, MR=1)  
LOAD(PLOTRC)  
REDUCE.  
EXECUTE.  
789 PUNCH  
SOURCE DECKS  
789 PUNCH  
DATA  
PLOT CARDS (SEE DESCRIPTION )  
6789 PUNCH (YELLOW CARD)
```

JOB CONTROL CARDS USING UPDATE FILE
WITHOUT PLOT TAPE

```
JOBN,CM72000,T4777.  
ATTACH(CONUPF, ID=JWE, PW=ADDY, MR=1)  
ATTACH(OVRLY, ID=JWE, MR=1)  
UPDATE(P=CONUPF)  
FTN(A,I=COMPILE,LR=0)  
SEGLOAD(I=OVRLY)  
LOAD(LGO)  
EXECUTE.  
789 PUNCH  
*IDENT NAME  
    UPDATES  
*COMPILE CONTROL.SUBSCL  
*END  
789 PUNCH  
    DATA  
6789 PUNCH      (YELLOW CARD)
```

JOB CONTROL CARDS USING UPDATE FILE
WITH PLOT TAPE

```
JCBN,CM72000,T4777,NT1.  
ATTACH(CONUPF, ID=JWE, PW=ADDY, MR=1)  
ATTACH(OVRLY, ID=JWE, MR=1)  
LABEL (CARD) SEE DESCRIPTION  
UPDATE(P=CONUPF)  
FTN(A,I=COMPILE,LR=0)  
SEGLOAD(I=OVRLY)  
LOAD(LGO)  
EXECUTE.  
EXIT.  
BKSP(INPUT)  
ATTACH(PLOTRC, ID=JWE, PW=ADDY, MR=1)  
LOAD(PLOTRC)  
REDUCE.  
EXECUTE.  
789 PUNCH  
*IDENT NAME  
    UPDATES  
*COMPILE CONTROL.SUBSCL  
*END  
789 PUNCH  
    DATA  
PLOT CARDS      (SEE DESCRIPTION )  
6789 PUNCH      (YELLOW CARD)
```

THE UPDATE FILE CAN BE USED TO MODIFY SOURCE ROUTINES IN THE CONTROL PROGRAM. UPDATES ARE USED TO INSERT OR DELETE CARDS OF A SPECIFIED SUBROUTINE. EXAMPLES OF THE UPDATE DIRECTIVES CAN BE FOUND IN THE UPDATE REFERENCE MANUAL. THE SOURCE LISTING FOR THE CONTROL PROGRAM IS CONTAINED IN ROOM 2115.

DESCRIPTION OF LABEL CARD
CC1
LABEL(TAPE6,W,D=HD,L=XXXXXXXXXX,VSN=YYYY)*RING IN*ZZZ**
XXXXXXXXXX=YOUR LABEL
YYYY=VOLUME NO. OBTAINED FROM COMPUTER ROOM
ZZZ=YOUR PAYROLL NUMBER

DESCRIPTION OF PLOT CARDS
PLOT CARDS
CARD ONE
CC1
PLOT
CARD TWO
CC1-4 CC 26-35 CC 41-44
NNNN NAME OF SUBTASK
SUBMITTER (4 DIGIT NUMBER)
CARD THREE-N
CC1-10 CC11-20
Y Y
MINIMUM MAXIMUM
(FLOATING POINT)
(ROOT LOCUS OR ROOT CONTOUR ONLY)
NNNN-VSN NUMBER

TABLE III. CONTROL: SYSTEM MODELS

A. CONTINUOUS SYSTEM MODELS

1. OPEN LOOP

$$\begin{aligned} C\dot{x} &= Ax + Bu \\ y &= Hx + G\dot{x} + Fu \end{aligned}$$

2. CLOSED LOOP

$$\begin{aligned} C\dot{x} &= Ax + Bu \\ u &= K_1 x + K_2 \dot{x} + D u_{com} \\ y &= Hx + G\dot{x} + Fu \end{aligned}$$

3. ROOT LOCUS

$$\begin{aligned} C\dot{x} &= Ax + Bu \\ u &= (K_1 x + K_2 \dot{x}) + (K_3 x + K_4 \dot{x}) \end{aligned}$$

B. DISCRETE SYSTEM MODELS

1. OPEN LOOP

$$\begin{aligned} x_{n+1} &= Ax_n + Bun \\ y_n &= Hx_n + Fun \end{aligned}$$

2. CLOSED LOOP

$$\begin{aligned} x_{n+1} &= Ax_n + Bun \\ u_n &= K_1 x_n + D u_{comm} \\ y_n &= Hx_n + Fun \end{aligned}$$

3. ROOT LOCUS

$$\begin{aligned} x_{n+1} &= Ax_n + Bun \\ u_n &= (K_1 x_n) + (K_3 x_n) \end{aligned}$$

C. SAMPLED-DATA SYSTEM MODELS

1. ORIGINAL SYSTEM

$$\begin{bmatrix} \dot{x}_n^c \\ x_{n+1}^d \end{bmatrix} = \begin{bmatrix} A_c & 0 \\ 0^* & A_d \end{bmatrix} \begin{bmatrix} x_n^c \\ x_n^d \end{bmatrix} + \begin{bmatrix} B_c & 0 \\ 0 & B_d \end{bmatrix} \begin{bmatrix} u_n^c \\ u_n^d \end{bmatrix}$$

$$\begin{bmatrix} y_n^c \\ y_n^d \end{bmatrix} = \begin{bmatrix} H_c & 0 \\ 0^* & H_d \end{bmatrix} \begin{bmatrix} x_n^c \\ x_n^d \end{bmatrix} + \begin{bmatrix} F_c & 0 \\ 0 & F_d \end{bmatrix} \begin{bmatrix} u_n^c \\ u_n^d \end{bmatrix}$$

2. CONNECTIONS PRIOR TO DISCRETIZATION

$$\begin{bmatrix} u_n^c \\ u_n^d \end{bmatrix} = [C] \begin{bmatrix} y_n^c \\ y_n^d \end{bmatrix}$$

WHERE C IS DEFINED BY YTOV AND ZTOU AND DEFINES CONNECTIONS MADE BEFORE THE PLANT IS DISCRETIZED (SEE TABLE VII B.)

3. DISCRETIZED SYSTEM

$$\begin{bmatrix} x_{n+1}^c \\ x_{n+1}^d \end{bmatrix} = \begin{bmatrix} \phi(t) & 0 \\ 0^* & A_d \end{bmatrix} \begin{bmatrix} x_n^c \\ x_n^d \end{bmatrix} + \begin{bmatrix} [\phi(t)B_c] & [\phi(t)B_d] & 0 \\ 0 & 0 & 0 & B_d \end{bmatrix} \begin{bmatrix} u_n^c \\ u_n^d \end{bmatrix}$$

$$\begin{bmatrix} y_n^c \\ y_n^d \end{bmatrix} = \begin{bmatrix} H_c & 0 \\ 0^* & H_d \end{bmatrix} \begin{bmatrix} x_n^c \\ x_n^d \end{bmatrix} + \begin{bmatrix} [F_c] & [H_cB_c] & 0 \\ 0 & 0 & 0 & F_d \end{bmatrix} \begin{bmatrix} u_n^c \\ u_n^d \end{bmatrix}$$

4. CONNECTIONS AFTER DISCRETIZATION

$$\begin{bmatrix} u_n^c \\ u_n^d \end{bmatrix} = [R] \begin{bmatrix} y_n^c \\ y_n^d \end{bmatrix} = [K_1] \begin{bmatrix} x_n^c \\ x_n^d \end{bmatrix}$$

WHERE R IS DEFINED BY YZTOK AND DEFINES CONNECTIONS MADE AFTER THE PLANT IS DISCRETIZED (SEE TABLE VII B.). FOR ROOT LOCUS, THE SECOND CONNECTION SPECIFIED BY YZTCK (IF ANY) WILL GENERATE K3 (SIMILAR TO K1) DEFINING A SECOND FEEDBACK VARIABLE.

5. FINAL SAMPLED-DATA SYSTEM

$$x_{n+1} = Ax_n + Bu_n \quad \text{FROM 3.}$$

$$y_n = Hx_n + Fu_n$$

$$u_n = K_1 x_n + D u_{\text{comm}} \quad \text{FROM 4. } (D=I)$$

* THESE SUBMATRICES MAY CONTAIN NON-ZERO ELEMENTS.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE IV CONTROL PROGRAM DATA DECK FORMAT

CARD 1 TITLE CARD FORMAT (10A8)
ANY LITERAL DATA DESIRED TO LABEL ALL PRINTOUTS AND PLOTS

CARD(S) 2 TO N NAMELIST CODE (SEE APPENDIX 2)
INTEGER VARIABLES
READ, SYSTFM, CPUTPUT, MIXED, DIGITL, FRPS, NUMERS, TRESP, NX, NY, NU, NXC,
NUC, ZOH, N1, N2, CCNTUR, MULTRT, MODEL, NSCALE, CMAT, NK2, FCRM, IPT, IGO,
SAV, IFLAG, READ3

REAL VARIABLES
DELT, FINALT, IFREQ, FFREQ, DELFRQ, GAIN1, GAIN2, M

CARD N+1 OUTPUT LABELS FFORMAT (8A10)
LITERAL OUTPUT VARIABLE LABELS USED TO LABEL PRINTOUTS AND PLOTS.
ORDERED IN SEQUENCE CORRESPONDING TO NY OUTPUTS, Y. LEAVE BLANK IF
SYSTEM = 3. Will read the greater of 8 or NY outputs.

CARD N+2 INPUT LABELS FFORMAT (8A10)
LITERAL INPUT VARIABLE LABELS USED TO LABEL PRINTOUTS AND PLOTS.
ORDERED IN SEQUENCE CORRESPONDING TO NU INPUTS, U. LEAVE BLANK IF
SYSTEM = 3. Will read the greater of 8 or NU inputs

CARD(S) (N+3) TO (M) SYSTEM DATA (SEE TABLE V AND APPENDIX 2)
SYSTEM DATA AS SPECIFIED BY LOAD, MATRIX, CHANGE, OR CLASS. IF READ = 1,
EACH DATA MATRIX IS READ ROW-WISE WITH FORMAT (8F10.4) AND EACH
MATRIX MUST BE PRECEDED BY A DIMENSION CARD (FORMAT (2I10)) GIVING
THE NUMBER OF ROWS AND COLUMNS OF THE MATRIX.

CARD M+1 TRANSIENT RESPONSE INPUT DATA
INPUT IS CALLED BY THIST TO GENERATE THE TRANSIENT RESPONSE INPUT
VECTOR. THE INPUT SUBROUTINE ON THE DISC READS ONE DATA CARD FOR
EACH RESPONSE DEFINING A STEP INPUT ON THE NU COMPONENTS OF THE
augmented and thinned input vector, u. (format(7F10.4))

THE ABOVE CARDS DEFINE ONE CASE. AS MANY CASES AS DESIRED MAY BE STACKED
TOGETHER FOR A SINGLE COMPUTER RUN.

IF A PLOT IS REQUESTED THE PLOT CARDS AS DESCRIBED IN TABLE II ARE
REQUIRED

TABLE V. DATA REQUIRED BY CONDITION CODES

STEP_1 READ = 1,2,3

SYSTEM	REQUIRED MATRICES
1 OPEN LOOP	A,B,C,H,G,F
IF MIXED =1, THIS IS STEP 1 OF THE MIXED LOADING OPTION	

2 CLOSED LOOP	A,B,C,H,G,F,K1,K2,D
3 RLOCUS	A,B,C,K1,K2,K3,K4

IF CMAT = 0	C MATRIX NOT REQUIRED
IF NK2 = 0	K2,K4 MATRIX NOT REQUIRED
IF OUTPUT = 1	G,F MATRIX NOT REQUIRED
IF OUTPUT = 2	F MATRIX NOT REQUIRED
IF OUTPUT = 3	G MATRIX NOT REQUIRED
IF N2=0	K3,K4 MATRIX NOT REQUIRED

STEP_2 READ = 4 ; OR,
IF MIXED =1, THIS IS STEP 2 OF THE MIXED LOADING OPTIONCARD 1 NBLOCK,NIT FORMAT (2I5)
IF NIT = 0 GO TO (*)

CARDS 2-(NBLOCK+1) ONE CARD PER BLOCK FORMAT (I2,I3,5I5,5F10.4)

NUM,TYPE,(CCNNEC(I),I=1,4),MOD,(PARAM(I),I=1,5)

NUM = BLOCK NUMBER

TYPE = BLOCK TYPE (1-10) SEE TABLE VI

CCNNEC = SPECIFIES INPUTS TO BLOCK. FIRST THREE ELEMENTS MAY SPECIFY
CONNECTIONS FROM OTHER BLOCKS (+). FOURTH ELEMENT MAY
SPECIFY EXTERNAL INPUT (+).MOD = SPECIFIES THAT G(P) IS S-,Z-, OR W- TRANSFORM As specified in A
PARAM = PARAMETERS DEFINING BLOCKS AS SPECIFIED IN TABLE VI

GO TO (**)

	NAME	TYPE	DIMENSION	FORMAT
(*)	GRAPH	INTEGER	NBLOCK X 5	5I5
	BLOCK	INTEGER	NBLOCK X 3	3I5
	NUMER	REAL	NBLOCK X 5	5F10.4
	DENOM	REAL	NBLOCK X 5	5F10.4
	GAIN	REAL	NBLOCK	8F10.4
(**)	I THIN	INTEGER	S # OF OUTPUTS	16I5

STOP IF MIXED =0 UNLESS READ=4 and SYSTEM =3

I THINL	INTEGER	S # OF INPUTS	16I5
NYTOV,NZTOU,NYZTOK	INTEGER		3I5
YTOV	INTEGER	NYTOV X 2	2I5
ZTOU	INTEGER	NZTOU X 2	2I5
YZTOK	INTEGER	NYZTOK X 2	2I5

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE VI TRANSFER FUNCTION STANDARD FORMS (NIT = 1)

FOR READ = 4 OR MIXED =1 STEP 2, THE CLASS SUBROUTINE ACCEPTS BLOCK DIAGRAM TRANSFER FUNCTIONS OF THE FOLLOWING FORMS. (SEE TABLE V. STEP2)

TYPE	G(P)	PARAM(I)				
		I=1	2	3	4	5
1	K	K				
2	K _P	K				
3	K $\frac{1}{P}$	K				
4	$\frac{K}{(1+P/a)}$	K	a			
5	$\frac{K(1+P/b)}{(1+P/a)}$	K	a	b		
6	$\frac{K_P}{(P+a)}$	K	a			
7	$\frac{K}{(1+P/a)(1+P/b)}$	K	a	b		
8	$\frac{K}{1 + \frac{2g}{\omega}P + P^2/\omega^2}$	K	ω	g		
9	$\frac{K(1 + \frac{2g_2}{\omega_2}P + P^2/\omega_2^2)}{(1 + \frac{2g_1}{\omega_1}P + P^2/\omega_1^2)}$	K	ω_1	g_1	ω_2	g_2
10	$\frac{K(1+P/a)}{(1 + \frac{2g}{\omega}P + P^2/\omega^2)}$	K	ω	g	a	
11	$\frac{K_P}{1 + 2\frac{g}{\omega}P + P^2/\omega^2}$ *	K	ω	g		

*Can consist of two real roots

TABLE VI (CONT.)

THE STANDARD FORM, $G(P)$, IS INTERPRETED AS AN S-, W-, OR Z-PLANE TRANSFER FUNCTION AS SPECIFIED BELOW

MOD DIGITL	0	1	2
0	$G(S)$	---	---
1	$G(S), G(Z)$	$G(S)$	$G(W)$
2	$G(Z)$	$G(S)$	$G(W)$

IF MOD = 0 THE TRANSFER FUNCTION COEFFICIENTS ARE NOT MODIFIED.

IF MOD = 1 THE TRANSFER FUNCTION CRITICAL FREQUENCIES ARE PREWARPED TO APPROXIMATE THE S-PLANE FILTER IN THE W- PLANE AND THEN TRANSFORMED TO THE Z-PLANE AS DESCRIBED BELOW. FIRST ORDER POLES AND ZEROES ARE WARPED AS

$$\alpha = \tanh\left(\frac{\alpha T}{2}\right)$$

COMPLEX POLES AND ZEROES ARE WARPED AS

$$\omega_w = \sqrt{u_w^2 + v_w^2} \quad z_w = -\frac{u_w}{\omega_w}$$

where $u_w = \frac{\sinh(\alpha T)}{\cosh(\alpha T) + \cos(\beta T)}$

$$v_w = \frac{\sin(\beta T)}{\cosh(\alpha T) + \cos(\beta T)}$$

with $\omega_s^2 = \alpha^2 + \beta^2$

$$z_s = -\frac{\alpha}{\omega_s}$$

IF MOD = 2 THE TRANSFER FUNCTION IS REGARDED AS A W-PLANE TRANSFER FUNCTION AND TRANSFORMED TO THE Z-PLANE BY THE BILINEAR TRANSFORMATION

$$h = (z-1)/(z+1)$$

IF DIGITL = 1 AND MOD $\neq 0$ THE TRANSFER FUNCTION CANNOT BE PART OF THE CONTINUOUS PLANT (I.E. THE INPUT TO THE FILTER MUST BE NUMBERED HIGHER THAN NUC AND THE FILTER STATES MUST BE NUMBERED HIGHER THAN NXc)

IF DIGITL = 1 AND MOD = 0 THE TRANSFER FUNCTION COEFFICIENTS ARE NOT MODIFIED. THE FUNCTION IS INTERPRETED AS AN S- OR Z- PLANE FUNCTION DEPENDING ON THE RELATION OF THE FILTER STATES TO NXc. IF THE FILTER STATES ARE NUMBERED LESS THEN NXc, THEN THE FILTER IS TREATED AS PART OF THE CONTINUOUS PLANT. IF THE FILTER STATES ARE NUMBERED GREATER THAN NXc THEN THE FILTER IS TREATED AS PART OF DIGITAL CONTROLLER.

TABLE VII DISCRETIZED PLANT MODELS FOR VARIOUS INPUT AND OUTPUT DEFINITIONS

	A. (*) $y_n = \bar{y}_n \equiv y(nT - \epsilon)$	B. (**) $y_n = \bar{y}_n^+ \equiv y(nT + \epsilon)$
CONTINUOUS MODEL	$\dot{x} = Ax + Bu$ $y = Hx + Fu$	
SAMPLED INPUT	$x_{n+1} = \phi(T)x_n + \phi(T)Bu_n$ $y_n = Hx_n$	$x_{n+1} = \phi(T)x_n + \phi(T)Bu_n$ $y_n = Hx_n + HBu_n \quad (F \equiv 0)$
ZOH INPUT	$x_{n+1} = \phi(T)x_n + \oplus(T)Bu_n$ $y_n = Hx_n$	$x_{n+1} = \phi(T)x_n + \oplus(T)Bu_n$ $y_n = Hx_n + Fu_n$

	C. (***) TIME DELAY $y_n(m) \equiv y[(n+m-1)T] \quad 0 < m \leq 1$
CONTINUOUS MODEL	$\dot{x} = Ax + Bu$ $y = Hx + Fu$
SAMPLED INPUT ($F \equiv 0$)	$x_{n+1} = \phi(T)x_n + \phi(T)Bu_n$ $y_n(m) = H\phi(mT)x_{n-1} + H\phi(mT)Bu_{n-1}$
ZOH INPUT	$x_{n+1} = \phi(T)x_n + \oplus(T)Bu_n$ $y_n(m) = H\phi(mT)x_{n-1} + [H\oplus(mT)B + F]u_{n-1}$

$$\phi(t) = \int_0^t e^{-A(t-\tau)} d\tau ; \quad \oplus(t) = \int_0^t \phi(t-\tau) d\tau ; \quad x_n = \bar{x}_n \equiv x(nT - \epsilon)$$

- (*) CONTROL ASSUMES $y_n = \bar{y}_n$ FOR ANY CONNECTIONS FROM THE PLANT TO THE DIGITAL CONTROLLER (I.E. PLANT FEEDBACK).
- (**) CONTROL ASSUMES $y_n = \bar{y}_n^+$ FOR FINAL SYSTEM OUTPUT CALCULATIONS (TRANSIENT RESPONSES, TRANSFER FUNCTIONS).
- (***) MODIFIED Z-TRANSFORM ANALYSIS CAN ONLY BE USED WITH OPEN LOOP SAMPLED-DATA SYSTEMS.

TABLE VIII VECTOR ORDERING AND SYSTEM CONNECTION CONVENTIONS FOR SAMPLED-DATA SYSTEMS

A. VECTOR ORDERING CONVENTION

THE COMPONENTS OF THE AUGMENTED VECTORS U, X, AND Y MUST BE ORDERED IN THE FOLLOWING SEQUENCES, (IN STEP 2 OF THE MIXED OPTION, NUMBER THE EXTERNAL INPUTS AND BLOCK OUTPUTS IN THE INDICATED ORDER)

INPUT VECTOR, U

1. INPUTS TO PLANT FROM ZERO-ORDER-HOLD ELEMENTS (ZOH)
2. INPUTS TO PLANT FROM SAMPLERS (NUC-ZOH)
3. INPUTS TO DIGITAL CONTROLLER (NU-NUC)

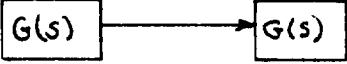
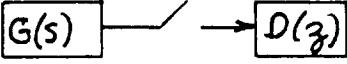
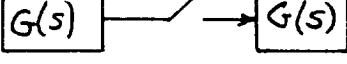
STATE VECTOR, X

1. CONTINUOUS STATES (PLANT, NXC)
2. DISCRETE STATES (DIGITAL CONTROLLER, NX-NXC)

OUTPUT VECTOR, Y

1. PLANT OUTPUTS
2. DIGITAL CONTROLLER OUTPUTS

B. SYSTEM CONNECTION CONVENTION

CONNECTION	DEFINED IN
	YTCV, ZTOU, GRAPH, A
	"
	"
	YZTOK
	"

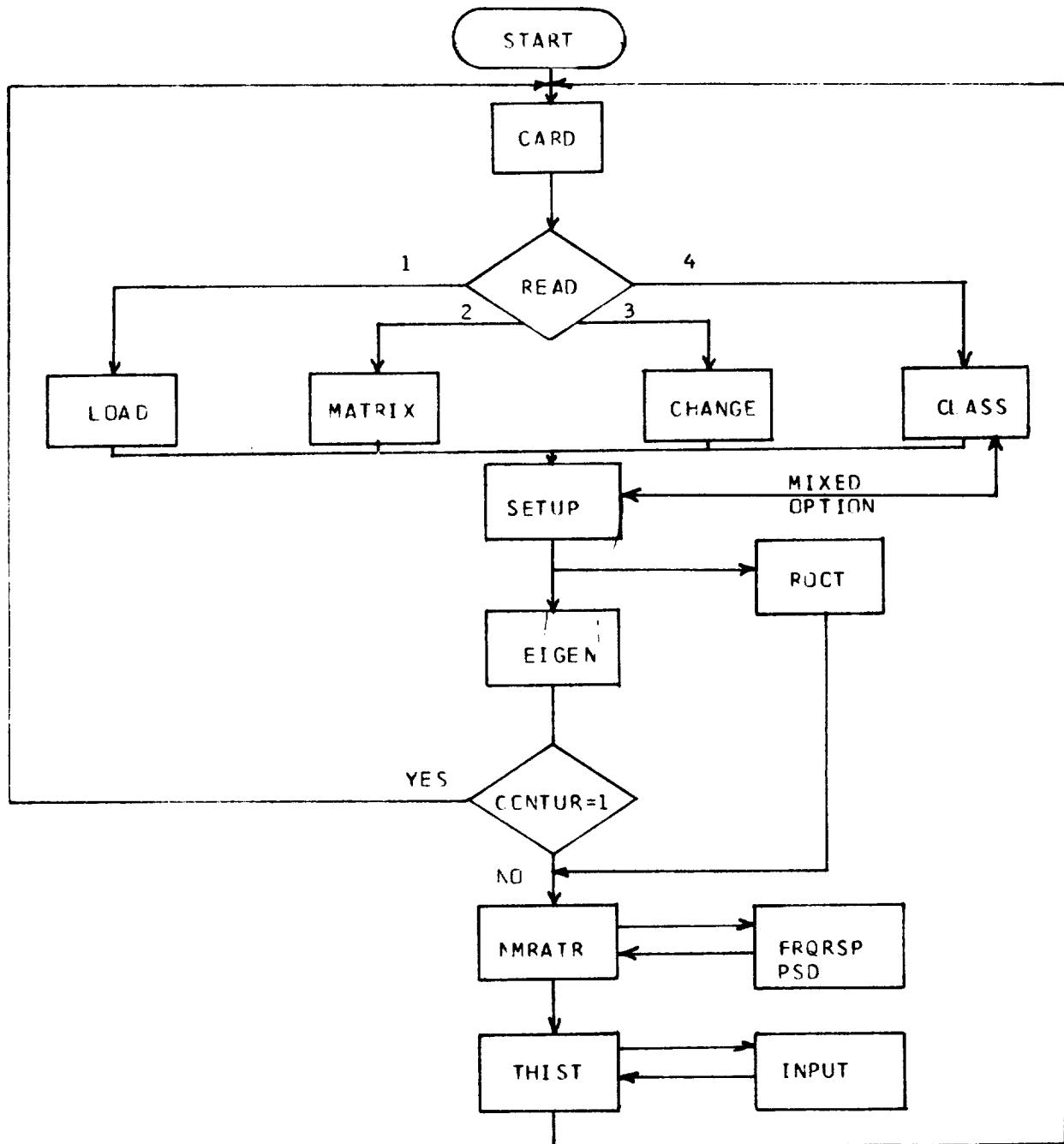
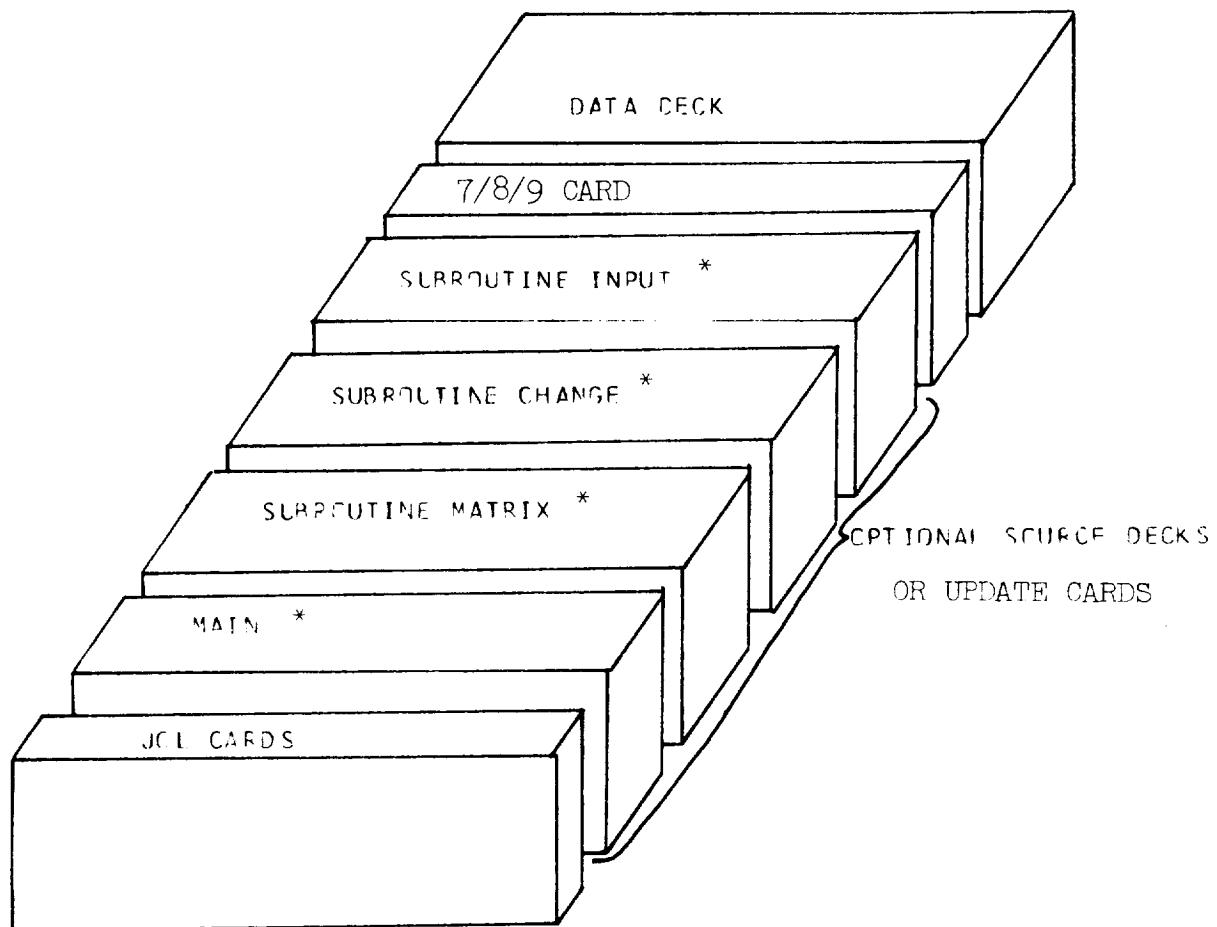


FIGURE 1 FLOW CHART OF CNTRLR SUBROUTINE



*OTHER SUBROUTINES AS DESIRED

FIGURE 2. STRUCTURE OF CONTROL DECK

ORIGINAL PAGE IS
OF POOR QUALITY

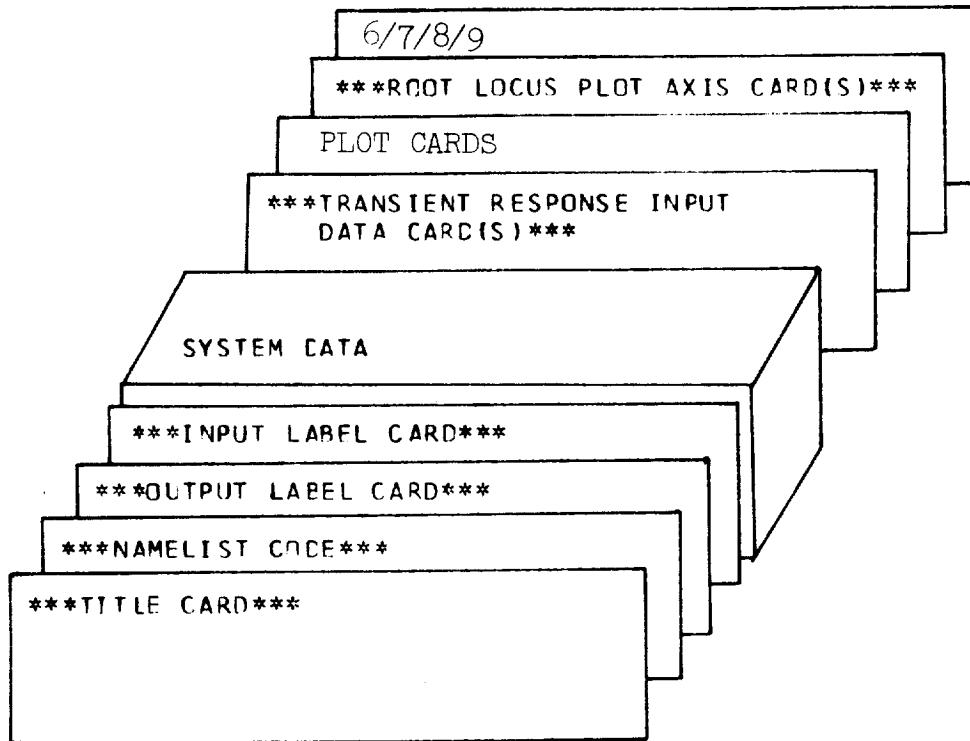


FIGURE 3. CONTROL PROGRAM DATA DECK STRUCTURE

```
CC2
|
$CODE      READ=1,      SYSTEM= 1,OUTPUT=3,NX=4,NY=5,NU=2,
           DELT=.05 ,FINAL T=10.,  TFREQ=.5 ,FFRFQ= 100.,DELFREQ=1.11,
           IPT=2,$END
```

FIGURE 4 EXAMPLE OF DATA ENTRY USING A NAMELIST FORMAT

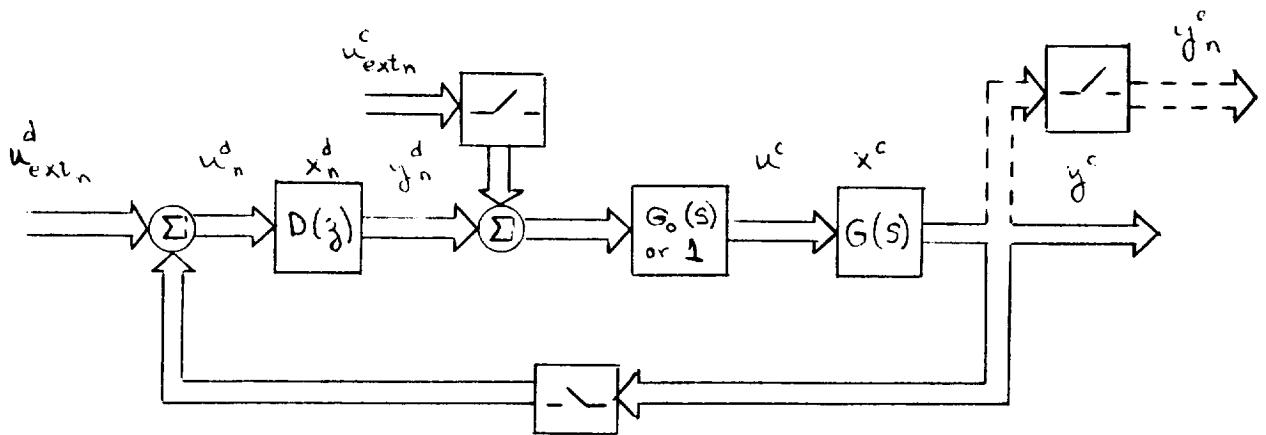


Figure 1.- Sampled-data system block diagram.

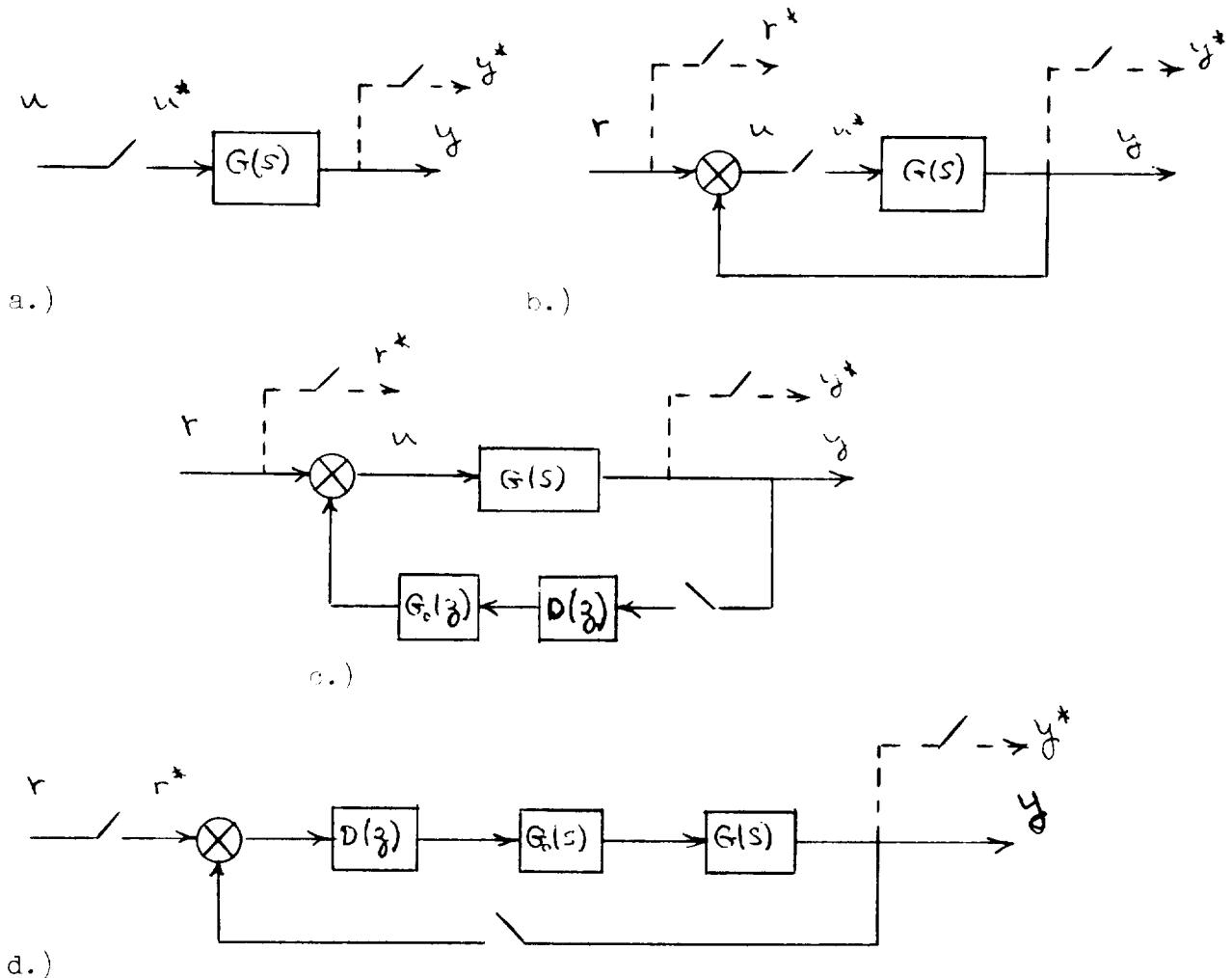


Figure 2.- Typical sampled-data systems analyzed by CONTROL.

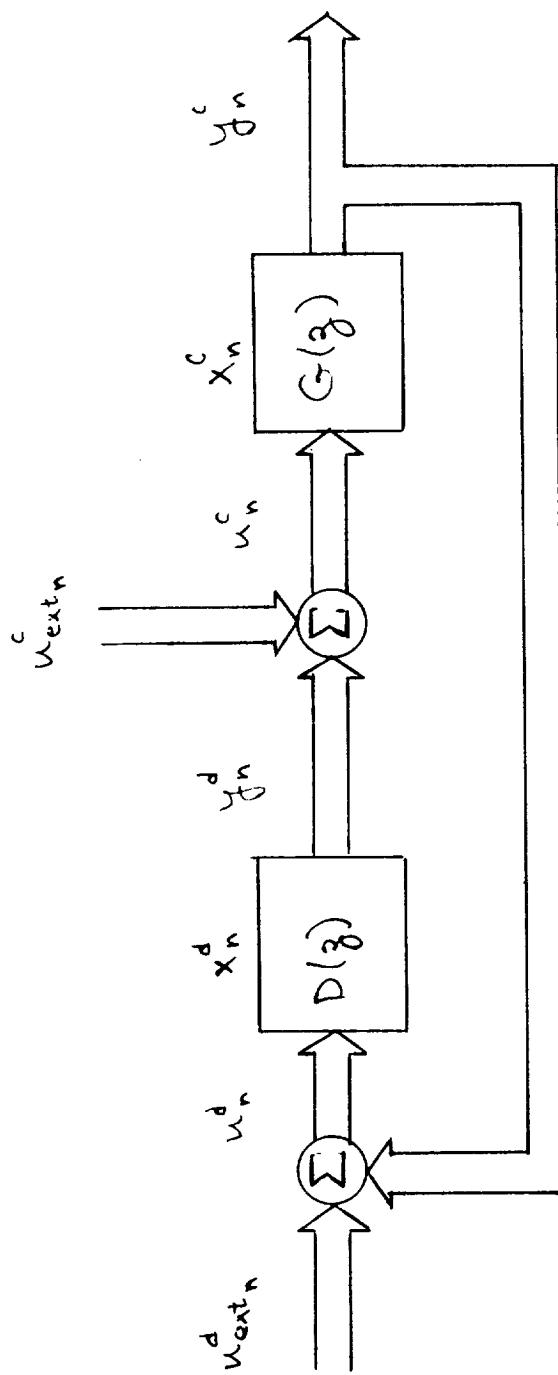
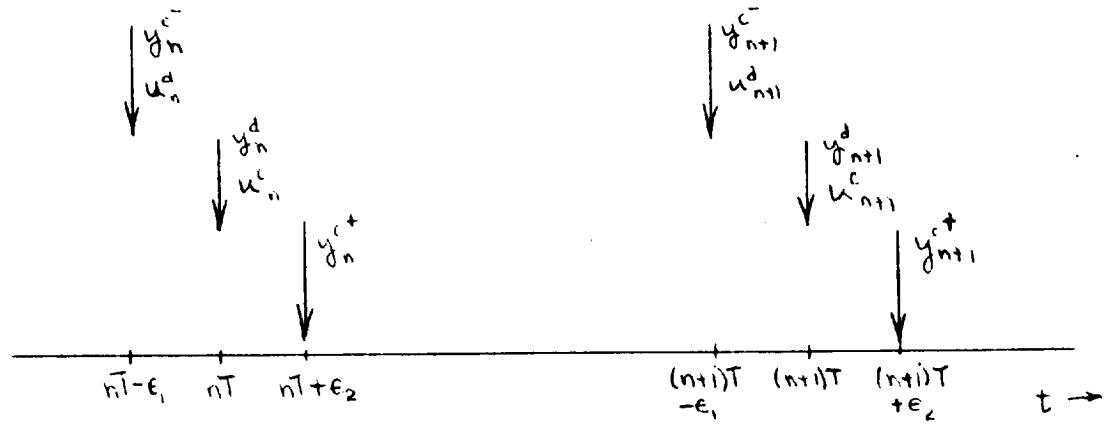


Figure 7.- Discretized sampled-data system block diagram.



a.) Time sequence of digital controller and plant.



b.) Idealized time sequence.

Figure 8.- Time sequence models for sampled-data systems.

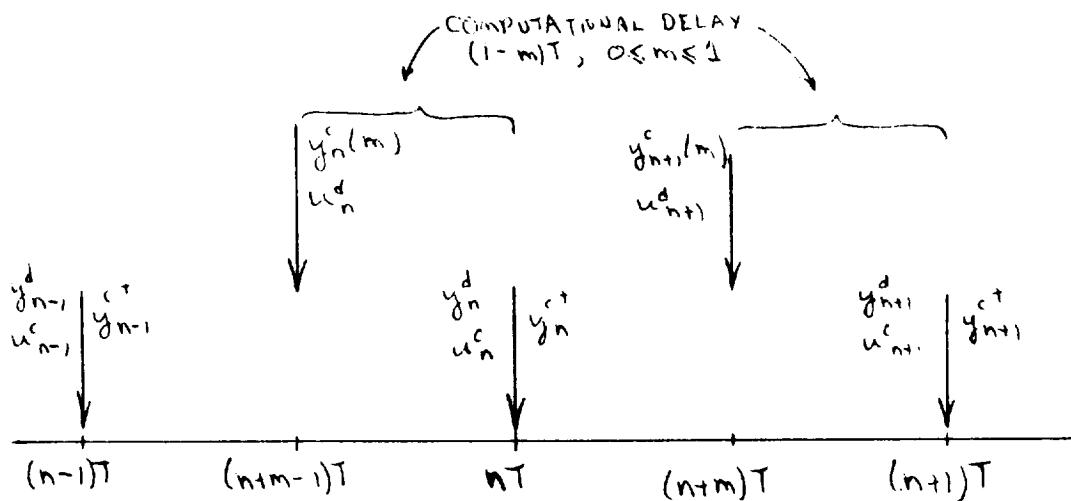


Figure 9.- Time sequence model for sampled-data systems with computational time delay.

(

)

)

)

)

(

